МАТЕРИАЛЫ ЗАДАНИЙ
Всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада»

по профилю «Искусственный интеллект»

2024/25 учебный год

ntcontest.ru

УДК 373.5.016:004.8
ББК 74.263.2
И86

Авторы:
С. Е. Бойченко, И. А. Воронцов, А. А. Гаврилюк, Е. Н. Горечин, Ю. М. Земцова, О. В. Зубков, С. В. Ладанова, И. Б. Мамай, А. В. Резников

И86 Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада». Учебно-методическое пособие Том 13 **Искусственный интеллект**
— М.: Ассоциация участников технологических кружков, 2025. — 151 с.

ISBN 978-5-908021-12-8

Данное пособие разработано коллективом авторов на основе опыта проведения всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада» в 2024/25 учебном году, а также многолетнего опыта проведения инженерных соревнований для школьников. В пособии собраны основные материалы, необходимые как для подготовки к олимпиаде, так и для углубления знаний и приобретения навыков решения инженерных задач.

В издании приведены варианты заданий по профилю Национальной технологической олимпиады за 2024/25 учебный год с ответами, подробными решениями и комментариями. Пособие адресовано учащимся 8–11 классов, абитуриентам, школьным учителям, наставникам и преподавателям учреждений дополнительного образования, центров молодежного и инновационного творчества и детских технопарков.

Методические материалы также могут быть полезны студентам и преподавателям направлений, относящихся к группам:
01.00.00 Математика и механика
02.00.00 Компьютерные и информационные науки
09.00.00 Информатика и вычислительная техника
10.00.00 Информационная безопасность

ISBN 978-5-908021-12-8
9 785908 021128 >

УДК 373.5.016:004.8
ББК 74.263.2

Оглавление

1. Введение

1.1. Национальная технологическая олимпиада

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» (далее — Олимпиада, НТО) проводится в соответствии с распоряжением Правительства Российской Федерации от 10.02.2022 № 211-р при координации Министерства науки и высшего образования Российской Федерации и при содействии Министерства просвещения Российской Федерации, Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, Министерства промышленности и торговли Российской Федерации, Ассоциации участников технологических кружков, Агентства стратегических инициатив по продвижению новых проектов, АНО «Россия — страна возможностей», АНО «Платформа Национальной технологической инициативы» и Российского движения детей и молодежи «Движение Первых».

Проектное управление Олимпиадой осуществляет структурное подразделение Национального исследовательского университета «Высшая школа экономики» — Центр Национальной технологической олимпиады. Организационный комитет по подготовке и проведению Национальной технологической олимпиады возглавляют первый заместитель Руководителя Администрации Президента Российской Федерации С. В. Кириенко и заместитель Председателя Правительства Российской Федерации Д. Н. Чернышенко.

Национальная технологическая олимпиада — это командная инженерная Олимпиада, позволяющая школьникам работать в самых передовых инженерных направлениях. Она базируется на опыте Олимпиады Кружкового движения НТИ и проводится с 2015 года, а с 2016 года входит в перечень Российского совета олимпиад школьников и дает победителям и призерам льготы при поступлении в университеты.

Всего заявки на участие в десятом юбилейном сезоне (2024–25 гг.) самых масштабных в России командных инженерных соревнованиях подали более 140 тысяч школьников. Общий охват олимпиады с 2015 года превысил 880 тысяч участников.

НТО способствует формированию профессиональной траектории школьников, увлеченных научно-техническим творчеством и помогает им:

- определить свой интерес в мире современных технологий;
- получить опыт решения комплексных инженерных задач;
- осознанно выбрать вуз для продолжения обучения и поступить в него на льготных условиях.

Кроме того, НТО позволяет каждому участнику познакомиться с перспективными направлениями технологического развития, ведущими экспертами и найти единомышленников.

Ценности НТО

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Олимпиада создает уникальное пространство, основанное на общих ценностях и смыслах, которыми делятся все участники процесса: школьники, студенты, организаторы, наставники и эксперты. В основе Олимпиады лежит представление о современном технологическом образовании как новом укладе жизни в быстро меняющемся мире. Эта модель предполагает:

- доступность качественного обучения для всех, кто стремится к знаниям;
- возможность непрерывного развития;
- совместное формирование среды, где гуманитарные знания и новые технологии взаимно усиливают друг друга.

Это — образ общества будущего, в котором участники Олимпиады оказываются уже сегодня.

Решать прикладные задачи, нацеленные на умножение общественного блага

В заданиях Олимпиады используются актуальные вызовы науки и технологий, адаптированные под уровень школьников. Они имеют прикладной характер и отражают реальные потребности общества, а системное и профессиональное решение подобных задач способствует развитию общего блага. Олимпиада предоставляет возможность попробовать себя в этом направлении уже сегодня и найти единомышленников.

Создавать, а не только потреблять

Стремление к созданию нового ценится выше потребления готового, а ориентация на общественную пользу — выше личной выгоды. Это не исключает заботу о собственных интересах, но подчеркивает: творчество приносит больше удовлетворения, чем пассивное потребление. Олимпиада — совместный труд организаторов, партнеров и участников, в котором важнее стремление решать общие задачи, чем критика чужих усилий.

Работать в команде

Командная работа рассматривается не только как эффективный способ достижения целей, но и как основа для формирования сообщества, объединенного общими ценностями. Команда помогает раскрыть индивидуальность каждого, при этом сохраняя уважение к другим. Такие горизонтальные связи необходимы для реализации амбициозных технологических проектов. Олимпиада способствует формированию подобного сообщества и приглашает к его созданию всех заинтересованных.

Осваивать и ответственно развивать новые технологии

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ, объединенные интересом к современным технологиям, стремлением к их пониманию и созданию нового. Возможности технологий постоянно расширяются, однако развитие должно сопровождаться ответственностью. Этика инженера и ученого предполагает осознание последствий своих решений. Главное правило — создавая новое, не навредить.

Играть честно и пробовать себя

Ценится честная победа, достигнутая в рамках установленных правил. Это предполагает отказ от списывания, давления и манипуляций. Честная игра означает уважение к себе, команде и соперникам. Олимпиада поддерживается как безопасное пространство, где каждый может пробовать новое, не опасаясь ошибок, и постепенно становиться сильнее и увереннее в себе.

Быть человеком

Соревнования — это сложный и эмоционально насыщенный процесс, в котором особенно важны порядочность, вежливость и чуткость. Эмпатия, уважение и забота делают участие полезным и комфортным. Высоко ценится бережное отношение к людям и их труду, отказ от токсичной критики и готовность нести ответственность за слова и поступки. Участие в общем деле помогает не только окружающим, но и самому человеку.

Организационная структура НТО

НТО — межпредметная олимпиада. Спектр соревновательных направлений (профилей НТО) сформирован на основе актуального технологического пакета и связан с решением современных проблем в различных технологических отраслях. С полным перечнем направлений (профилей) можно ознакомиться на сайте НТО: https://ntcontest.ru/tracks/nto-school/.

Соревнования в рамках НТО проводятся по четырем трекам:

1. НТО Junior для школьников (5–7 классы).
2. НТО школьников (8–11 классы).
3. НТО студентов.
4. Конкурс цифровых портфолио «Талант НТО».

В 2024/25 учебном году 21 профиль НТО включен в Перечень олимпиад школьников, ежегодно утверждаемый Приказом Министерства науки и высшего образования Российской Федерации, а также в Перечень олимпиад и иных интеллектуальных и (или) творческих конкурсов, утверждаемый приказом Министерства просвещения Российской Федерации. Это дает право победителям и призерам профилей НТО поступать в вузы страны без вступительных испытаний (БВИ), получить 100 баллов ЕГЭ или дополнительные 10 баллов за индивидуальные достижения. Преимущества при поступлении победителям и призерам НТО предлагают более 100 российских вузов.

НТО для школьников 8–11 классов проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. Участникам предлагаются предметный тур, состоящий из задач по двум предметам, связанным

с выбранным профилем, а также инженерный тур, задания которого погружают участников в тематику профиля; образовательный модуль формирует теоретические знания и представления.
- Второй отборочный этап — заочный командный. На этом этапе участники выполняют как индивидуальные задания на проверку компетенций, так и командные задачи, соответствующие выбранному профилю.
- Заключительный этап — очный командный. В течение 5–6 дней команды участников со всей страны, успешно прошедшие оба отборочных этапа, соревнуются в решении комплексных прикладных инженерных задач.

Профили НТО 2024/25 учебного года и соответствующий уровень РСОШ

Профили II уровня РСОШ:

- Автоматизация бизнес-процессов.
- Автономные транспортные системы.
- Беспилотные авиационные системы.
- Водные робототехнические системы.
- Инженерные биологические системы.
- Наносистемы и наноинженерия.
- Нейротехнологии и когнитивные науки.
- Технологии беспроводной связи.
- Цифровые технологии в архитектуре.
- Ядерные технологии.

Профили III уровня РСОШ:

- Анализ космических снимков и геопространственных данных.
- Аэрокосмические системы.
- Большие данные и машинное обучение.
- Геномное редактирование.
- Интеллектуальные робототехнические системы.
- Интеллектуальные энергетические системы.
- Информационная безопасность.
- Искусственный интеллект.
- Летающая робототехника.
- Спутниковые системы.
- Кластер «Виртуальные миры»:
  - ◇ Разработка компьютерных игр.
  - ◇ Технологии виртуальной реальности.
  - ◇ Технологии дополненной реальности.

Профили без уровня РСОШ:

- Инфохимия.
- Квантовый инжиниринг.
- Новые материалы.
- Программная инженерия в финансовых технологиях.

- Современная пищевая инженерия.
- Умный город.
- Урбанистика.
- Цифровые сенсорные системы.
- Разработка мобильных приложений.

Обратите внимание на то, что в олимпиаде 2025/26 учебного года список профилей, в т. ч. входящих в РСОШ, и уровни РСОШ могут поменяться.

Участие в НТО старшеклассников может принять любой школьник, обучающийся в 8–11 классе. Чаще всего Олимпиада привлекает:

- учащихся технологических кружков, интересующихся инженерными и робототехническими соревнованиями;
- школьников, увлеченных олимпиадами и предпочитающих межпредметный подход;
- энтузиастов передовых технологий;
- активных участников хакатонов, проектных конкурсов и профильных школ;
- будущих предпринимателей, ищущих команду для реализации стартап-идей;
- любознательных школьников, стремящихся выйти за рамки школьной программы.

Познакомить школьников с НТО и ее направлениями, а также мотивировать их на участие в Олимпиаде можно с помощью специальных мероприятий — Урока НТО и Дней НТО. Методические рекомендации для педагогов по проведению Урока НТО и организации Дня НТО в образовательной организации размещены на сайте: https://nti-lesson.ru. Здесь можно подобрать и скачать готовые сценарии занятий и подборки материалов по различным направлениям Олимпиады.

Участвуя в НТО, школьники получают возможность работать с практико-ориентированными задачами в области прорывных технологий, собирать команды единомышленников, погружаться в профессиональное сообщество, а также заработать льготы для поступления в вузы.

По всей стране работают площадки подготовки к НТО, которые помогают привлекать участников и проводят мероприятия по подготовке к этапам Олимпиады. Такие площадки могут быть открыты на базе:

- школ и учреждений дополнительного образования;
- частных кружков по программированию, робототехнике и другим технологическим направлениям;
- вузов;
- технопарков и других образовательных и научно-технических организаций.

Любое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к Олимпиаде и присоединиться к Кружковому движению НТИ. Подробные инструкции о том, как стать площадкой подготовки, размещены на сайте: https://ntcontest.ru. Условия регистрации и требования к ним актуализируются с развитием Олимпиады, а обновленная информация публикуется перед началом каждого нового цикла.

Наставники НТО

В Национальной технологической олимпиаде большое внимание уделяется работе с **наставниками** — людьми, сопровождающими участников на всех этапах подготовки и участия в Олимпиаде. Наставник оказывает поддержку как в решении организационных вопросов, так и в развитии технических и социальных навыков школьников, включая умение работать в команде.

Наставником НТО может стать любой взрослый, готовый помогать школьникам развиваться и готовиться к участию в инженерных соревнованиях. Это может быть:

- учитель школы или преподаватель вуза;
- педагог дополнительного образования;
- руководитель кружка;
- родитель школьника;
- специалист из технологической области или представитель бизнеса.

Даже если наставник сам не обладает достаточными знаниями в определенной области, он может привлекать к подготовке коллег и экспертов, а также оказывать поддержку и организовывать процесс обучения для самостоятельных учеников. Сегодня сообщество наставников НТО насчитывает более **7 000 человек** по всей стране.

Главная цель наставника — **организовать системную подготовку к Олимпиаде в течение всего учебного года**, поддерживать интерес и мотивацию участников, а также помочь им справляться с возникающими трудностями. Также наставник фиксирует цели команды и каждого участника, чтобы в дальнейшем можно было проанализировать развитие профессиональных и личных компетенций.

Основные направления работы наставника

Организационные задачи:

- Информирование и мотивация: наставник рассказывает учащимся об НТО, ее этапах и преимуществах, помогает с выбором подходящего профиля, ориентируясь на интересы и способности школьников.
- Составление программы подготовки: формируется расписание и план занятий, организуется работа по освоению необходимых знаний и навыков.
- Контроль сроков: наставник следит за календарем Олимпиады и напоминает участникам о сроках решения заданий отборочных этапов.

Содержательная подготовка:

- Оценка компетенций участников: наставник помогает определить сильные и слабые стороны учеников и подбирает задания и материалы для устранения пробелов.
- Подготовка к отборочным этапам: помощь в изучении рекомендованных материалов, заданий прошлых лет, онлайн-курсы по профилям.
- Подготовка к заключительному этапу: разбираются задачи заключительных этапов прошлых лет, отслеживаются подготовительные мероприятия (очные и дистанционные), в которых наставник рекомендует ученикам участвовать.

Развитие личных и командных навыков:

- Формирование команд: наставник помогает сформировать сбалансированные команды для второго отборочного и финального этапов, распределить роли, при необходимости ищет участников из других регионов и организует онлайн-коммуникацию.
- Анализ прогресса и опыта: после каждого этапа проводится совместная рефлексия, обсуждаются успехи и трудности, выявляются зоны роста и направления для дальнейшего развития.
- Поддержка и мотивация: наставник поддерживает интерес и энтузиазм участников (особенно в случае неудачных результатов), помогает справиться с разочарованием и сохранить настрой на дальнейшее участие.
- Построение индивидуальной образовательной траектории: наставник помогает школьникам осознанно планировать дальнейшее обучение: выбирать курсы, участвовать в конкурсах, определяться с вузами и направлениями подготовки.

Поддержка наставников НТО

Работе наставников посвящен отдельный раздел на сайте НТО: https://ntcontest.ru/mentors/.

Для систематизации знаний и подходов к работе наставников в рамках инженерных соревнований разработан курс «Дао начинающего наставника: как сопровождать инженерные команды»: https://stepik.org/course/124633/. Курс формирует общие представления об их работе в области подготовки участников к инженерным соревнованиям.

Для совершенствования профессиональных компетенций по направлениям профилей создан курс «Дао начинающего наставника: как развивать технологические компетенции»: https://stepik.org/course/186928/.

Для организации занятий с учениками педагогам предлагаются образовательные программы, разработанные на основе многолетнего опыта организации подготовки к НТО. В настоящий момент они представлены по передовым технологическим направлениям:

- компьютерное зрение;
- геномное редактирование;
- водная, летающая и интеллектуальная робототехника;
- машинное обучение и искусственный интеллект;
- нейротехнологии;
- беспроводная связь, дополненная реальность.

Программы доступны на сайте: https://ntcontest.ru/mentors/education-programs/.

Регистрируясь на платформе НТО, наставники получают доступ к личному кабинету, в котором отображается расписание отборочных соревнований и мероприятий по подготовке, требования к знаниям и компетенциям при решении задач отборочных этапов.

Сообщество наставников НТО существует и развивается. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: https://konkurs.kruzhok.org/. Принять участие в конкурсе может каждый наставник.

В 2022 году было выпущено пособие «Технологическая подготовка инженерных команд. Методические рекомендации для наставников». Методические рекомендации предназначены для учителей технологий, а также наставников и педагогов кружков и центров дополнительного образования. Рекомендации направлены на помощь в процессе преподавания технологий в школе или в кружке. Пособие построено на примерах из реального опыта работы со школьниками, состоит из теоретических положений, посвященных популярным взглядам в педагогике на тему подготовки инженерных команд к соревнованиям. Электронное издание доступно по ссылке: https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih.

В нем рассмотрены особенности подготовки к пяти направлениям:

- Большие данные.
- Машинное обучение.
- Искусственный интеллект.
- Спутниковые системы.
- Летающая робототехника.

Для наставников НТО разработана и постоянно пополняется страница с материалами для профессионального развития: https://nto-forever.notion.site/c9b9cbd21542479b97a3fa562d15e32a.

1.2. Искусственный интеллект

Цель профиля — развитие у школьников прикладных навыков в сфере искусственного интеллекта через знакомство с методами машинного и глубокого обучения для решения задач, актуальных для науки и бизнеса.

Соревнования по профилю способствуют повышению уровня обеспечения российского рынка технологий искусственного интеллекта квалифицированными кадрами за счет:

- возрастания привлекательности конкурсов и олимпиад, направленных на развитие интеллектуальных и творческих способностей обучающихся (в соответствии с Национальной стратегией развития искусственного интеллекта Российской Федерации);
- вовлечения их в сферу искусственного интеллекта;
- ориентации школьников на профессиональное развитие в этой области.

В 2024/25 учебном году участникам Национальной технологической олимпиады по профилю Искусственный интеллект предложено погрузиться в задачу, направленную на сохранение биоразнообразия и защиту редких видов животных. В центре внимания проблема мониторинга популяции амурских тигров — вида, занесенного в Красную книгу России. Финалисты разрабатывают алгоритмы для реидентификации конкретных особей по изображениям с фотоловушек, что позволяет автоматизировать один из самых трудоемких процессов полевой зоологии.

Поставленная цель требует не только глубоких знаний в области компьютерного зрения и машинного обучения, но и внимательного отношения к качеству данных, умения работать с изображениями в сложных условиях и выстраивать точные и устойчивые модели.

Основная задача отборочных этапов Олимпиады данного профиля — выявить наиболее способных к решению подобных задач школьников и через образовательную составляющую развить у школьников прикладные навыки в области машинного обучения.

Первый отборочный (индивидуальный) дистанционный этап состоит из предметного тура по математике и информатике, а также инженерного тура, в рамках которого участникам предлагается решить базовую задачу машинного обучения, направленную на автоматизацию обработки данных с фотоловушек. Конкретно: они разрабатывают модель для классификации изображений диких животных по группам, организованным по биологическому сходству (например, все виды оленей рассматривались как одна группа). Условия приближены к реальным: фотографии были получены в разное время суток, в сложных природных условиях, с частичным попаданием животного в кадр или перекрытием объектов.

Целью задания является не только обучение базовой модели классификации, но и развитие навыков:

- обработки изображений;
- работы с нестандартными датасетами;
- выбора архитектур и метрик.

В целом, инженерный тур проверяет практические навыки работы с алгоритмами машинного обучения, способность к аналитическому мышлению, понимание предметной области и умение применять технические решения для реальных задач в сфере экологии и охраны природы.

Предметный тур определяет уровень подготовки школьников по предметам: математика и информатика (программирование).

Задания по информатике относятся к разделам: алгоритмы, программирование и методы оптимизации. Здесь школьники должны продемонстрировать простейшие навыки составления и отладки программ, обрабатывающих массивы данных, и понимание таких тем, как комбинаторика, операции со строками, матричный анализ, теория графов.

Задачи по математике проверяют у участников знания по алгебре, комбинаторике, теории вероятности и математической статистике.

Таким образом, задачи предметного тура выявляют знания, необходимые для решения задач следующего (второго) и заключительного этапов.

Дополнительно для участников был проведен онлайн-интенсив AI-ARROW, подразумевающий быстрое погружение в сферу ИИ, начиная с основ программирования на Python с использованием профильных библиотек и заканчивая подходами к разработке нейросетей с использованием технологий компьютерного зрения и обработки естественного языка.

В рамках второго отборочного этапа участникам предстояло решить две задачи — индивидуальную и командную.

Индивидуальная задача направлена на построение модели многоклассовой детекции животных на фотографиях с фотоловушек. Здесь нужно создать алгоритм, способный не только определить наличие животных в кадре, но и точно указать координаты ограничивающих прямоугольников для каждого из них, классифицируя по группам (например, медведь, куньи, пантеры и др.). Усложняют дело реальные условия съемки: перекрытия объектов, ночные кадры, частично видимые животные и разнообразие фонов. Метрикой оценки выступает mAP 0.5–0.95, учитывающая качество детекции при разных порогах перекрытия объектов.

Командная задача связана с классификацией животных по видам, а именно, требуется разработать модель, способную по фотографии определить конкретный вид животного (например, тигр, ирбис, манул, марал и др.), несмотря на внешние сходства между представителями разных видов и качество снимков. Работа в команде позволяет участникам эффективно распределить роли, комбинируя навыки в области обработки изображений, построения архитектур нейросетей и анализа ошибок модели.

Обе задачи второго этапа направлены не только на отбор сильнейших участников, но и на развитие ключевых компетенций в области компьютерного зрения, проектного мышления и командной инженерной работы.

Для подготовки финалистов дополнительно проводится образовательный хакатон, во время которого им предоставляется возможность погрузиться в процесс разработки проекта в области компьютерного зрения. Цель мероприятия заключается в создании модели, способной выполнять сегментацию животных на фотографиях с фотоловушек — то есть точно выделять каждую отдельную особь класса animal на изображении. Такой подход повышает точность учета животных, минимизирует ошибки, вызванные наложением объектов или сложными условиями съемки. В рамках хакатона финалисты оттачивают ключевые навыки, необходимые для успешного решения задачи заключительного этапа, а участники, не прошедшие в финал, — попробовать свои силы в решении задач более высокого уровня и продолжить развиваться в сообществе Академии ИИ.

Заключительный этап НТО состоит из командного инженерного и индивидуального предметного туров.

Предметный тур проводится по двум предметам — информатика и математика — и проверяет знания, необходимые в решении задач на анализ данных. Баллы, набранные в индивидуальном предметном туре (то есть знания и умения решать олимпиадные задачи в важных для направления разделах математики и информатики) влияют на личные результаты участников.

Задача командного инженерного тура заключительного этапа НТО 2024/25 посвящаются реидентификации амурских тигров — редкого и охраняемого вида, занесенного в Красную книгу Российской Федерации. Участники разрабатывают модель, способную определять конкретных особей по фотографиям с фотоловушек. Это имеет важное прикладное значение: помогает зоологам вести мониторинг популяции, отслеживать передвижение животных и разрабатывать меры по их охране. Для решения необходимо учитывать ряд факторов — от различий в узоре шкур тигров с разных сторон до частичного перекрытия объектов в кадре и нестандартных условий съемки. Исключительно ручная идентификация занимает много времени и требует экспертных знаний, тогда как применение технологий искусственного интеллекта позволяет существенно ускорить этот процесс.

Участникам важно продемонстрировать не только техническую компетентность в области компьютерного зрения, но и умение работать с «грязными» полевыми данными, интегрируя знания, полученные на предыдущих этапах. Работа требует высокой слаженности внутри команды — от выбора архитектуры и стратегии обучения до точной настройки модели и подготовки финального решения.

Таким образом, учащиеся 8–11 классов, прошедшие все этапы НТО, демонстрируют понимание основных операций, необходимых для построения модели машинного обучения, а именно:

- подготовка и анализ данных с целью выявления признаков, необходимых для решения поставленной задачи;
- построение модели;
- интерпретация результатов ее работы для оценки качества в рамках специализированной доменной области (машинное зрения для распознавания животных).

Для того чтобы сделать это возможным, в ходе Олимпиады проводится цикл образовательных и отборочных мероприятий. С момента регистрации для подготовки участникам доступны:

- Курс по машинному обучению (основной уровень) от Академии ИИ: https://ai-academy.ru/training/courses/kurs-po-mashinnomu-obucheniyu-osnovnoj-uroven/.
- Курс по машинному обучению (продвинутый уровень) от Академии ИИ: https://ai-academy.ru/training/courses/kurs-po-mashinnomu-obucheniyu-prodvinutij-uroven/.

- Материалы Академии искусственного интеллекта для школьников: https://ai-academy.ru/.

2. Первый отборочный этап

2.1. Работа наставника НТО на этапе

Педагог-наставник играет важную роль в подготовке участника к первому отборочному этапу Национальной технологической олимпиады. На этом этапе школьникам предстоит справиться как с предметными задачами, соответствующими профилю, так и с заданиями инженерного тура, погружающими в выбранную технологическую область.

Наставник может организовать подготовку участника, используя разнообразные форматы и ресурсы:

- Разбор заданий прошлых лет. Совместный анализ задач отборочного этапа предыдущих лет позволяет понять структуру, уровень сложности и типичные подходы к решению. Это формирует у школьника устойчивые стратегии работы с олимпиадными заданиями.
- Мини-соревнования. Проведение тренировочных турниров с заданиями предметных олимпиад муниципального уровня помогает развить соревновательный навык, тренирует скорость и уверенность при решении задач в ограниченное время.
- Углубленные занятия. Наставник может выстроить образовательную траекторию, опираясь на рекомендации разработчиков профиля, и провести занятия по ключевым темам. Это особенно важно для системного понимания предметной области.
- Использование онлайн-курсов. Для самостоятельной подготовки и проверки знаний участник может использовать предметные курсы НТО, размещенные на платформах Степик и Яндекс Контест. Наставник может также организовать занятия с использованием этих материалов в рамках групповой или индивидуальной подготовки.
- Привлечение внешних экспертов. Если у наставника нет достаточной экспертизы в какой-либо предметной области, он может пригласить других педагогов или специалистов для проведения тематических занятий.
- Поддержка в инженерном туре. Инженерный тур включает теоретические материалы и задания, помогающие глубже погрузиться в тематику профиля. Наставник может сопровождать изучение курса, помогать в разборе теоретических вопросов и тренировать участника на практических задачах.

Таким образом, наставник не только помогает систематизировать подготовку, но и мотивирует участника, создавая для него комфортную и продуктивную образовательную среду.

2.2. Предметный тур. Информатика

2.2.1. Первая волна. Задачи 8–11 класса

Задачи первой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63452/enter/.

Задача 2.2.1.1. Ускорение ускорения (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим модель движения тела. Будем фиксировать такие параметры, как координата, скорость, ускорение и ускорение ускорения (рывок). Если некоторый параметр равен a и имеет скорость изменения v, то в следующий момент времени этот параметр будет равен $a + v$.

Например, если тело имело координату, равную 10, скорость, равную 20, ускорение, равное 30 и ускорение ускорения, равное 40, то в следующий момент оно будет иметь координату 30, скорость 50 и ускорение 70. Ускорение ускорения будем считать в этой задаче постоянной величиной.

Задача довольно проста: тело в начальный момент времени 0 находится в точке с координатой 0, скоростью 0 и ускорением 0. На это тело действует постоянное ускорение ускорения, равное 6. Требуется определить, в точке с какой координатой окажется это тело в момент времени t.

Формат входных данных

В единственной строке находится одно число t, где $0 \leqslant t \leqslant 10^6$.

Формат выходных данных

Вывести одно число — координату, в которой окажется тело в момент времени t.

Примеры

Пример №1

Стандартный ввод
6

Стандартный вывод
120

Пример №2

Стандартный ввод
2

Стандартный вывод
0

Пример №3

Стандартный ввод
1000000

Стандартный вывод
999997000002000000

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int t;
    cin >> t;
    cout << ((t * (t - 1)) * (t - 2)) << endl;
}
```

Задача 2.2.1.2. Двойное остекление (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

У деда Василия есть два прямоугольных куска стекла. Один из них имеет размеры $a \times b$, другой — $c \times d$. Дед собирается из этих кусков сделать окно с двойным остеклением. Он хочет, чтобы окно было обязательно квадратным и как можно большим по размеру. Дед должен вырезать из имеющихся у него прямоугольников два одинаковых квадрата максимально возможного размера. Нужно написать программу, которая по заданным a, b, c, d найдет максимальные размеры квадратного окна. Имейте ввиду, что оба квадрата могут быть вырезаны и из одного прямоугольного куска стекла.

Формат входных данных

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника a, b через пробел, во второй — размеры второго прямоугольника c, d через пробел, где $1 \leqslant a, b, c, d \leqslant 10^9$.

Формат выходных данных

Вывести одно число — максимальную сторону квадратного двойного окна, которое можно вырезать из заданных на входе прямоугольных кусков стекла. Ответ может быть нецелым, требуется вывести его с точностью 1 знак после десятичной точки.

Примеры

Пример №1

Стандартный ввод

```
5 10
9 6
```

Стандартный вывод

```
5
```

Пример №2

Стандартный ввод

```
4 10
9 6
```

Стандартный вывод

```
4.5
```

Комментарий

Второй пример показывает, что иногда лучше вырезать оба квадрата из одного и того же куска стекла.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    double a, b, c, d;
    cin >> a >> b >> c >> d;
    double a0 = min({a, b, c, d});
    double a1 = min(max(a, b) / 2.0, min(a, b));
    double a2 = min(max(c, d) / 2.0, min(c, d));
    double ans =  max({a0, a1, a2});
    if( (int)ans == ans ){
        int ians = ans;
        cout << ians << endl;
        return 0;
    }
    cout.precision(1);
    cout << fixed<< ans << endl;
}
```

Задача 2.2.1.3. О золотой рыбке и... досках (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

После событий известной сказки А. С. Пушкина старик решил принципиально не пользоваться услугами золотой рыбки. Поэтому для того чтобы изготовить новое корыто, он честно заготовил n одинаковых досок.

Но гостивший в это время у старика со старухой внук решил, что ему нужно научиться пилить. И, не сказав ничего своему деду, внук быстро распилил каждую из досок на две части. В итоге у старика оказались $2n$ кусков досок. Самое интересное, что все эти куски оказались разными по длине, но имели целочисленные размеры. К сожалению, старик забыл, какова была исходная длина целых досок.

Формат входных данных

В первой строке задается целое число n — исходное количество целых досок, где $1 \leqslant n \leqslant 10^5$.

Во второй строке заданы $2n$ целых чисел d_i — длины всех кусков, которые получились после «тренировки» внука, где $1 \leqslant d_i \leqslant 10^9$. Гарантируется, что эти числа попарно различны, и их можно разбить на пары одинаковых по сумме чисел.

Все эти части досок пронумерованы от 1 до $2n$ в том порядке, в котором они заданы на входе.

Формат выходных данных

В первую строку вывести одно число — исходную длину целых досок.

В следующих n строках вывести пары номеров кусков досок, которые составляют по длине целые доски. Номера выводить через один пробел, внутри пары сначала должен идти меньший номер, затем больший. Пары должны быть выведены в порядке возрастания первых номеров в парах.

Примеры

Пример №1

Стандартный ввод
3 4 8 2 3 6 7

Стандартный вывод
10 1 5 2 3 4 6

Комментарий

Отсортируем куски и далее будем брать один из начала и второй к нему из конца.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    vector<pair<int, int> > v(2 * n);
     for(int i = 0; i < 2 * n; i++){
        int d;
        cin >> d;
        v[i] = {d, i + 1};
    }
    sort(v.begin(), v.end());
    vector<pair<int, int> > ans(n);
    for(int i = 0; i < n; i++){
```

```
            ans[i] = {v[i].second, v[2 * n - i - 1].second};
            if(ans[i].first > ans[i].second){
                swap(ans[i].first, ans[i].second);
            }
        }
        sort(ans.begin(), ans.end());
        cout << v[0].first + v.back().first<< endl;
        for(int i = 0; i < n; i++){
            cout << ans[i].first<<' '<< ans[i].second<< endl;
        }
    }
```

Задача 2.2.1.4. Бонусы и экономия (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Технология производства некоторой металлической детали предполагает вытачивание ее из металлической заготовки. При этом образуются стружки, которые не стоит выкидывать. Ведь из a комплектов стружек (оставшихся после обработки a заготовок) можно бесплатно выплавить еще одну заготовку, которую снова можно использовать для выточки детали и создания еще одного комплекта стружек.

Заготовки можно купить на оптовом складе, при этом в целях привлечения клиентов, проводится акция «купи b заготовок, тогда еще одну получишь бесплатно».

Требуется изготовить c деталей. Нужно определить минимальное число заготовок, которые нужно купить за деньги, чтобы с учетом бонусных заготовок и экономии на стружках можно было изготовить требуемое число деталей.

Формат входных данных

В одной строке через пробел заданы три целых числа a, b, и c такие, что $2 \leqslant a \leqslant 10^{18}$, $1 \leqslant b$, $c \leqslant 10^{18}$.

Формат выходных данных

Вывести одно целое число — минимальное количество заготовок, которые нужно купить, чтобы с учетом всех бонусов и экономии выточить c конечных деталей.

Примеры

Пример №1

Стандартный ввод
`4 5 41`

Стандартный вывод
`26`

Примечания

В примере из условия нужно закупить 26 заготовок. Тогда за каждые пять купленных заготовок будет предоставлена одна бесплатная, итого по акции добавится еще пять заготовок, то есть получится 31 заготовка. Далее из 31 заготовки выточится 31 деталь, останется 31 комплект стружек. Из каждых четырех комплектов выплавится дополнительная заготовка, получится семь заготовок и три комплекта стружек. Из семи заготовок выточится семь деталей и останется семь комплектов стружек, три комплекта стружек осталось с первого шага, итого 10 комплектов стружек. Из них выплавится еще две заготовки, дающие две детали и два комплекта стружек. Собрав эти два комплекта с двумя, оставшимися от 10, получим еще одну заготовку, из которой выточится еще одна деталь. Останется один комплект стружек, который уже никак не получится использовать. Итого будет произведена $31 + 7 + 2 + 1 = 41$ деталь.

Комментарий

Методом бинарного поиска можно подобрать минимальное необходимое количество исходных заготовок.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
int f1(int M, int a){
    int res = 0, z = 0;
    while(1){
        if(M == 0 && z < a){
            return res;
        }
        res += M;
        M = M + z;
        z = M % a;
        M = M / a;
    }
}
```

```
int f2(int M, int b){
    return M + M / b;
}
signed main(){
    int a, b, c;
    cin >> a >> b >> c;
    int L = 0, R = 1;
    while(f1(R, a) <= c){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f1(M, a) < c){
            L = M;
        }
        else{
            R = M;
        }
    }
    int z = R;
    L = 0, R = 1;
    while(f2(R, b) <= z){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f2(M, b) < z){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << R << endl;
}
```

Задача 2.2.1.5. Сон таксиста (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одному таксисту приснился красочный сон. Во сне он живет и работает в некотором городе, где абсолютно все улицы с односторонним движением. Эти улицы устроены так, что невозможно проехать с какого-либо перекрестка так, чтобы вернуться обратно на этот же перекресток, то есть в дорожной сети города нет циклов.

Таким образом, если с перекрестка A можно попасть по направлению движения улиц на перекресток B, то люди вызывают такси, иначе их везет специальный муниципальный подземный транспорт бесплатно.

В связи с такими странными правилами, таксистам в этом городе разрешено законом везти пассажира по любому маршруту, не нарушающему направления движения. Все в этом городе привыкли к такой ситуации и абсолютно спокойно относятся к тому, что таксисты везут их самым длинным путем. Разумеется, заработок таксиста за одну поездку прямо пропорционален ее длине. Для упрощения будем считать, что стоимость 1 км поездки составляет ровно 1 руб.

Схема дорог города задана. Перекрестки города пронумерованы числами от 1 до n. Таксист в своем сне находится на перекрестке номер S. Напишите программу, которая подскажет ему, сколько он максимально сможет заработать, когда ему придет заказ от клиента. Так как он не знает, куда попросит его везти клиент, нужно для каждого перекрестка от 1 до n указать максимальную стоимость поездки до этого перекрестка из пункта S на такси. Если по правилам на такси добраться из пункта S до какого-то перекрестка нельзя, вывести -1.

Формат входных данных

Дорожная сеть задана следующим образом: в первой строке находятся два числа через пробел n и m — число перекрестков и число улиц в городе, где $2 \leqslant n, m \leqslant 2 \cdot 10^5$.

В следующих m строках задана очередная односторонняя улица в виде трех чисел A, B, d через пробел, где A — начало улицы, B — конец улицы и d — ее длина. $1 \leqslant A, B \leqslant n$, $1 \leqslant d \leqslant 10^9$. Гарантируется, что в этой дорожной сети нет циклов. Некоторые пары перекрестков могут быть соединены двумя и более односторонними улицами. Дорожная сеть может быть неплоской за счет мостов и тоннелей.

В последней строке ввода содержится номер стартового перекрестка S, $1 \leqslant S \leqslant n$.

Формат выходных данных

Вывести n чисел в одну строку через пробел. i-е число обозначает длину самого длинного пути с перекрестка номер S до перекрестка номер i. Если до перекрестка номер i от S нельзя доехать, не нарушая правила движения, вывести -1.

Примеры

Пример №1

Стандартный ввод
10 20 9 10 15 9 8 3 8 10 7 7 8 4 7 10 10 5 8 2 5 9 10

Стандартный ввод
5 6 5 7 6 5 4 6 8 3 6 4 3 4 6 5 3 2 2 5 2 2 3 3 3 1 5 1 4 2 2 1 7 4 7 4 6 8 1 5

Стандартный вывод
7 -1 2 9 0 18 13 19 10 26

Комментарий

Задача решается методом динамического программирования на ориентированном ациклическом графе.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int n, m;
vector<vector<pair<int, int> > > G;
vector<int> order, used;
void dfs(int a){
    used[a] = 1;
    for(auto to : G[a]){
        if(!used[to.first]){
            dfs(to.first);
        }
    }
    order.push_back(a);
}
signed main(){
    cin >> n >> m;
    G.resize(n + 1);
    used.resize(n + 1, 0);
    for(int i = 0; i < m; i++){
        int a, b, d;
        cin >> a >> b >> d;
        G[a].push_back({b, d});
    }
```

```
    int s;
    cin >> s;
    dfs(s);
    reverse(order.begin(), order.end());
    vector<int> dp(n + 1, -1);
    dp[s] = 0;
    for(auto el : order){
        for(auto to : G[el]){
            dp[to.first] = max(dp[to.first], dp[el] + to.second);
        }
    }
    for(int i = 1; i <= n; i++){
        cout << dp[i] <<' ';
    }
}
```

2.2.2. Вторая волна. Задачи 8–11 класса

Задачи второй волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63454/enter/.

Задача 2.2.2.1. Игра на планшете (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Маленький Андрей изучает геометрические фигуры при помощи игры на планшете. У него есть прямоугольные треугольники четырех цветов и ориентаций: желтые, зеленые, красные и синие. Для каждой разновидности треугольников есть заданное количество экземпляров этих треугольников. Более точно: у Андрея есть a желтых, b зеленых, c красных и d синих треугольников. Помимо этого у него есть прямоугольная таблица $n \times m$.

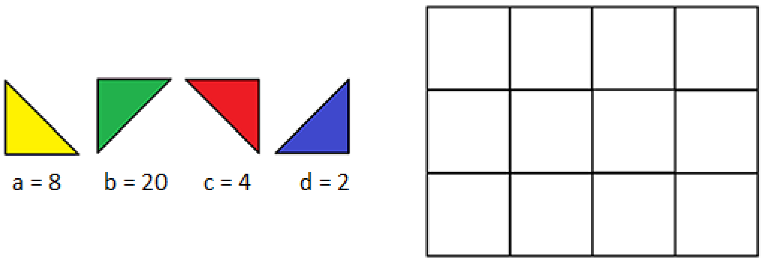


Рис. 2.2.1

Треугольники одного цвета имеют одну и ту же ориентацию, которую нельзя поменять. Андрей может только взять очередной треугольник и переместить его параллельным сдвигом в одну из ячеек этой прямоугольной таблицы. При этом в одну ячейку можно поместить либо вместе желтый и красный треугольники, либо вместе зеленый и синий, либо один любой треугольник из имеющихся.

Андрей хочет расположить в ячейках таблицы как можно больше треугольников из тех, что у него имеются. Нужно подсказать ему максимальное количество треугольников, которые получится разместить в таблице.

Формат входных данных

В первой строке содержатся четыре целых числа a, b, c и d через пробел — количество желтых, зеленых, красных и синих треугольников соответственно.

Во второй строке содержатся два целых числа n и m через пробел — размеры прямоугольной таблицы.

Все числа в пределах от 1 до 10^9.

Формат выходных данных

Вывести одно число — максимальное количество треугольников, которые можно при заданных условиях разместить в таблице.

Примеры

Пример №1

Стандартный ввод
`8 20 4 2` `3 4`

Стандартный вывод
`18`

Примечания

На рис. 2.2.2 представлен один из примеров размещения 18 треугольников из 34 заданных на входе.

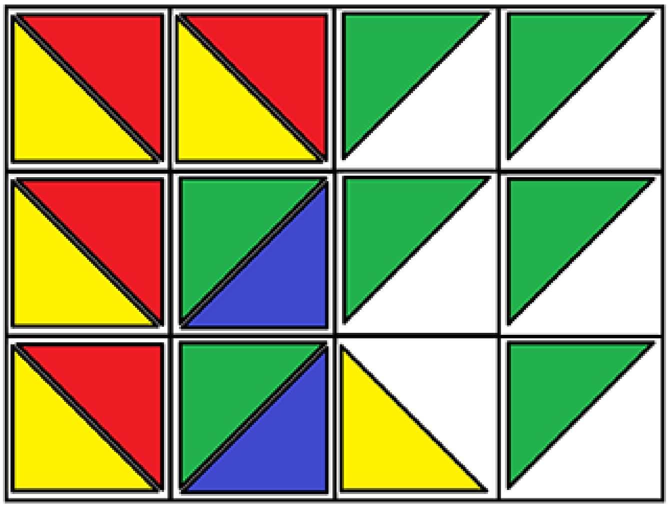


Рис. 2.2.2

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d, n, m;
    cin >> a >> b >> c >> d >> n >> m;
    if(a > c){
        swap(a, c);
    }
    if(b > d){
        swap(b, d);
    }
    int f = a + b;
    int k = n * m;
    if(k <= f){
        cout << k * 2;
        return 0;
    }
    k -= f;
    c -= a;
    d -= b;
    cout << f * 2 + min(k, c + d) << endl;
}
```

Задача 2.2.2.2. Старая задача на новый лад (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одна старая задача имеет следующий вид:

«Разбить число 45 на сумму четырех слагаемых так, что если к первому прибавить 2, из второго вычесть 2, третье умножить на 2, а четвертое разделить на 2, то получится одно и то же число».

Ответ к этой задаче — четыре числа 8, 12, 5 и 20. Можно убедиться, что в сумме они дают число 45, а если с каждым из них проделать соответствующую арифметическую операцию, то получится одно и то же число 10.

Необходимо решить чуть более общую задачу: даны числа n и k. Нужно представить число n в виде суммы четырех целых неотрицательных слагаемых $a + b + c + d$ таких, что $a + k = b - k = c \cdot k = d/k$. Гарантируется, что для заданных n и k такое разбиение существует.

Формат входных данных

В одной строке через пробел два числа n и k, где $1 \leqslant n \cdot k \leqslant 10^{18}$.

Формат выходных данных

Вывести через пробел в одну строку четыре целых неотрицательных числа a, b, c, d таких, что $a + b + c + d = n$ и $a + k = b - k = c \cdot k = d\,/\,k$.

Примеры

Пример №1

Стандартный ввод
`45 2`

Стандартный вывод
`8 12 5 20`

Пример №2

Стандартный ввод
`128 7`

Стандартный вывод
`7 21 2 98`

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n, k;
    cin >> n >> k;
    int x = (k * n) / (k * k + 2 * k + 1);
    cout << x - k <<' '<< x + k <<' '<< x / k <<' '<< x * k << endl;
}
```

Задача 2.2.2.3. Ладья и обязательная клетка (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Шахматная ладья находится в левом верхнем углу прямоугольного поля, разбитого на клетки размером $n \times m$. n обозначает число строк, m — число столбцов. Она хочет попасть в правую нижнюю клетку этого поля кратчайшим путем. Ладья может передвигаться либо вправо, либо вниз на любое количество клеток. Ладья обязана посетить заданную клетку с координатами (x, y), где x — номер строки этой клетки, а y — номер ее столбца.

Требуется найти количество способов построить путь ладьи из левого верхнего угла в правый нижний, которые проходят через обязательную клетку с заданными координатами.

Формат входных данных

В первой строке находятся два числа через пробел: n — число строк и m — число столбцов прямоугольного поля, $2 \leqslant n,\ m \leqslant 25$. Во второй строке через пробел находятся координаты (x, y) обязательной для посещения клетки, где $1 \leqslant x \leqslant n$, $1 \leqslant y \leqslant m$. Координаты x и y не совпадают с координатами левой верхней и правой нижней клеток.

Формат выходных данных

Вывести одно число — количество кратчайших путей ладьи из верхней левой в правую нижнюю клетку, проходящих через заданную клетку.

Примеры

Стандартный ввод
3 4 2 3
Стандартный вывод
6

Примечания

На рис. 2.2.3 представлены шесть путей, которыми ладья может пройти по полю размером 3×4, обязательно посещая по пути клетку $(2, 3)$.

Комментарий

Задачу можно решить как комбинаторными методами (произведение биномиальных коэффициентов), так и динамическим программированием.

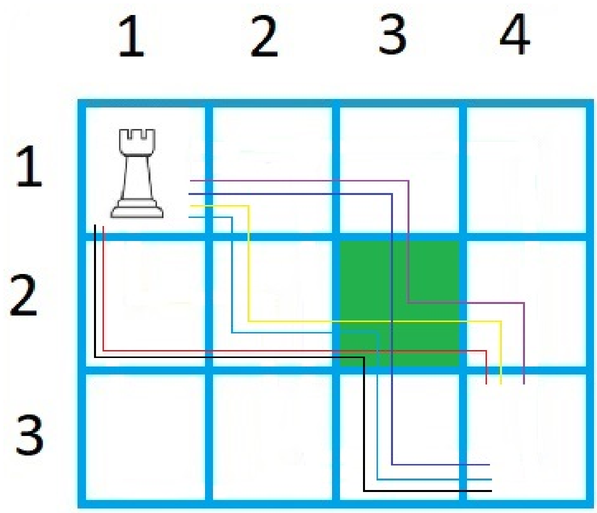


Рис. 2.2.3

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    vector<vector<int> > bc(51, vector<int>(51, 0));
    bc[0][0] = 1;
    for(int i = 1; i <= 50; i++){
        for(int j = 0; j < 51; j++){
```

```
            bc[i][j] += bc[i - 1][j];
            if(j - 1 >= 0){
                bc[i][j] += bc[i - 1][j - 1];
            }
        }
    }
    int n, m, x, y;
    cin >> n >> m >> x >> y;
    int d1 = bc[x - 1 + y - 1][x - 1];
    int d2 = bc[n - x + m - y][n - x];
    int ans = d1 * d2;
    cout << ans << endl;
}
```

Задача 2.2.2.4. Танец с цифрами (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Десять танцоров репетируют на сцене новый танец. Каждый танцор одет в футболку, на которой написана одна из цифр от 1 до 9, цифры могут повторяться. Изначально они стоят в некотором порядке слева направо, и их цифры образуют некоторое десятизначное число A. Далее во время всего танца участники либо разбиваются на пять пар рядом стоящих танцоров и одновременно меняются местами внутри своих пар, либо самый левый танцор перемещается на самую правую позицию и становится самым правым танцором.

Сын постановщика танца от скуки на бумаге выписывает все получающиеся при каждом перемещении десятизначные числа. Так как танец длинный, то в итоге на бумаге окажутся все возможные числа, которые в принципе могут появится при этих условиях. Нужно найти разницу между самым большим и самым маленьким из этих чисел.

Формат входных данных

На вход подается одно десятизначное число A, обозначающее начальное расположение танцоров. В числе могут встречаться цифры от 1 до 9, некоторые из них могут повторяться.

Формат выходных данных

Вывести одно число, равное разности самого большого и самого маленького из чисел, которые могут быть получены во время танца.

Примеры

Пример №1

Стандартный ввод
1456531355

Стандартный вывод
5182160085

Примечания

Самое маленькое число, которое можно получить в примере, равно 1353155456, самое большое равно 6535315541.

Покажем, как получить эти числа из исходного числа 1456531355. Сначала получим самое большое следующим образом: две левых цифры, 1 и 4, переместим вправо, получим 5653135514, потом поменяем в парах цифры местами и получим самое большое — 6535315541. Далее опять поменяем порядок в парах и в числе 5653135514 переместим три левых цифры 5, 6 и 5 вправо, получим 3135514565 и здесь снова поменяем порядок в парах, получим самое маленькое — 1353155456. Таким образом, искомая разница равна 5182160085.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    string s;
    cin >> s;
    string mx = s, mn = s;

    for(int i = 0; i < 5; i++){
        for(int j = 0; j < 10; j++){
            mx = max(mx, s);
            mn = min(s, mn);
            if(j < 9){
                s = s.substr(1) + s[0];
            }
        }
        for(int j = 0; j < 5; j++){
            swap(s[2 * j], s[2 * j + 1]);
        }
    }
    stringstream ssmn;
    ssmn << mn;
    int imn;
    ssmn >> imn;
    stringstream ssmx;
```

```
    ssmx << mx;
    int imx;
    ssmx >> imx;
    cout << imx - imn << endl;
}
```

Задача 2.2.2.5. Трудная сортировка (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

Иннокентий работает в отделе сортировки перестановок, подотделе сортировки вставками. Его задача заключается в сортировке перестановок, предоставленных заказчиками. Перестановкой длины n называется такая последовательность чисел, в которой встречаются все числа от 1 до n без повторений в некотором порядке.

Перестановка считается отсортированной, если в ней все числа расположены по возрастанию, то есть она имеет вид $1, \ldots, n$.

Иннокентий начинает рабочий день с пустой последовательности чисел. За день он сортирует вставками перестановку длины n. В начале каждой операции вставки он получает очередное число a_i из перестановки заказчика, после чего обрабатывает его, вставляя в отсортированную последовательность из ранее полученных чисел. После каждого такого добавления последовательность уже обработанных чисел должна быть отсортирована по возрастанию.

Перед тем как вставить число a_i в последовательность, он может выбрать, с какого края последовательности начать вставку. Далее он устанавливает число a_i с этого края и последовательно меняет вставляемое число с рядом стоящим числом b_j до тех пор, пока число a_i не встанет на свое место. На каждую перестановку вставляемого числа a_i с числом b_j Иннокентий тратит b_j единиц энергии.

Дана перестановка длины n из чисел a_i в том порядке, в котором Иннокентий их будет обрабатывать. Подскажите ему, какое минимальное количество энергии ему потребуется потратить, чтобы отсортировать всю перестановку.

Формат входных данных

В первой строке находится одно целое число n — длина перестановки, где $1 \leqslant n \leqslant 2 \cdot 10^5$.

Во второй строке содержится n целых чисел a_i через пробел в том порядке, в котором они поступают на обработку Иннокентию. Гарантируется, что эти числа образуют перестановку длины n, то есть каждое число от 1 до n содержится в заданном наборе ровно один раз.

Формат выходных данных

Вывести одно число — минимальные суммарные энергозатраты Иннокентия для сортировки вставками заданной на входе перестановки.

Примеры

Пример №1

Стандартный ввод
9 2 9 1 5 6 4 3 8 7

Стандартный вывод
43

Примечания

Первым устанавливается число 2. Оно ни с чем не меняется местами, поэтому затрат нет.

Далее устанавливается число 9. Выбираем правый край и ставим его туда без потерь энергии.

Затем устанавливаем число 1. Выбираем левый край, ставим его туда и снова потерь нет.

Теперь нужно вставить число 5. Если его вставлять с правого края, придется менять местами с 9, а если с левого, то с 1 и 2, что суммарно явно лучше. Итого затраты на вставку 5 равны 3.

Число 6 снова лучше вставить слева, затраты на его вставку равны 8.

Число 4 вставим слева за 3.

Число 3 так же слева за 3.

А вот число 8 лучше вставить справа за 9.

И осталось число 7. Если вставлять слева, то затратим 21, а если справа, то всего 17.

Итого на сортировку заданной перестановки потратили: $0 + 0 + 0 + 3 + 8 + 3 + 3 + 9 + 17 = 43$.

Комментарий

Построим дерево отрезков на сумму, при обработке числа a будем находить, какая сумма на данный момент меньше: от 1 до $a-1$ или от $a+1$ до n. Прибавим ее к ответу и поместим в позицию a это число a.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int LG = 19;
int N = (1 << LG);
vector<int> tr(2 * N, 0);
void upd(int pos, int x){
    pos += N;
    tr[pos] = x;
    pos /= 2;
    while(pos){
        tr[pos] = {tr[2 * pos]+ tr[2 * pos + 1]};
        pos /= 2;
    }
}
int get(int l, int r){
    l += N;
    r += N;
    int res = 0;
    while(l <= r){
        if(l % 2 == 1){
            res += tr[l];
        }
        if(r % 2 == 0){
            res += tr[r];
        }
        l = (l + 1) / 2;
        r = (r - 1) / 2;
    }
    return res;
}
signed main(){
    int n, a;
    cin >> n;
    int ans = 0;
    for(int i = 0; i < n; i++){
        cin >> a;
        int sl = get(0, a - 1);
        int sr = get(a + 1, N - 1);
        ans += min(sl, sr);
        upd(a, a);
    }
    cout << ans << endl;
}
```

2.2.3. Третья волна. Задачи 8–11 класса

Задачи третьей волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63456/enter/.

Задача 2.2.3.1. Туннель (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим классическую задачу прохождения группы с одним фонариком по туннелю. Есть четыре человека, и у них есть один фонарик. Нужно перевести всю группу на другой конец туннеля. По туннелю можно проходить только с фонариком и только либо вдвоем, либо в одиночку. По этой причине придется сделать пять рейсов по туннелю: три рейса туда и два рейса обратно. Туда идут двое, обратно — один, возвращая фонарик еще не прошедшей части группы. У каждого из четырех человек своя скорость передвижения по туннелю, но некоторые скорости могут совпадать. Двое идут со скоростью самого медленного в этой паре. Нужно найти минимальное время, за которое можно перевести группу по туннелю.

Здесь, в зависимости от скоростей персонажей, есть две стратегии. Проиллюстрируем их на примерах.

Пусть есть люди A, B, C, D. У A — время прохождения туннеля 1 мин, у B — 4 мин, у C — 5 мин, у D — 10 мин. Здесь работает наиболее очевидная стратегия: самый быстрый переводит текущего и возвращается с фонариком обратно за следующим. При этой стратегии нужно проходить так:

- A, B туда, затрачено 4 мин;
- A обратно, затрачена 1 мин;
- A, C туда, затрачено 5 мин;
- A обратно, затрачена 1 мин;
- A, D туда, затрачено 10 мин.

Общее время $4 + 1 + 5 + 1 + 10 = 21$ мин.

Но не всегда эта стратегия оптимальна. Уменьшим время прохождения туннеля персонажем B до 2 мин. По вышеопределенной стратегии будет 19 мин ($2 + 1 + 5 + 1 + 10 = 19$), но имеется более быстрое решение:

- A, B туда, затрачено 2 мин;
- A обратно, затрачена 1 мин;
- C, D туда, затрачено 10 мин;
- B обратно, затрачено 2 мин;
- A, B туда, затрачено 2 мин.

Общее время $2 + 1 + 10 + 2 + 2 = 17$ мин.

Заметим, что для предыдущего примера такая стратегия не работает: $4+1+10+4+4 = 23$ мин.

Если же персонаж B проходит туннель за 3 мин (а все остальные так же, как и в примерах), то независимо от стратегии будет затрачено 20 мин. В этом случае считаем, что работает первая стратегия.

Поразмыслив, станет понятно, от какого условия зависит выбор стратегии. Далее будем всегда считать, что A движется не медленнее B, B движется не медленнее C, C движется не медленнее D.

Дано время прохождения туннеля персонажами A, C, D. Нужно найти границу `border` для B такую, что если определить для B время прохождения строго меньшее, чем `border`, то выгодна вторая стратегия, иначе — первая.

Формат входных данных

В одной строке задано три целых чисел через пробел — время прохождения туннеля персонажами A, C, D. Времена даны по неубыванию. Все числа на входе в пределах от 1 до 100.

Формат выходных данных

Вывести одно число — границу `border` для B такую, что если определить время прохождения им туннеля строго меньше, чем `border`, нужно использовать вторую стратегию, иначе — первую. Ответ может быть нецелым, поэтому вывести его нужно с одним знаком после десятичной точки.

Примеры

Пример №1

Стандартный ввод
1 5 10

Стандартный вывод
3

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int A, C, D;
    cin >> A >> C >> D;
    cout.precision(1);
    cout << fixed << (A + C) / 2.0 << endl;
}
```

Задача 2.2.3.2. Математический пазл (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

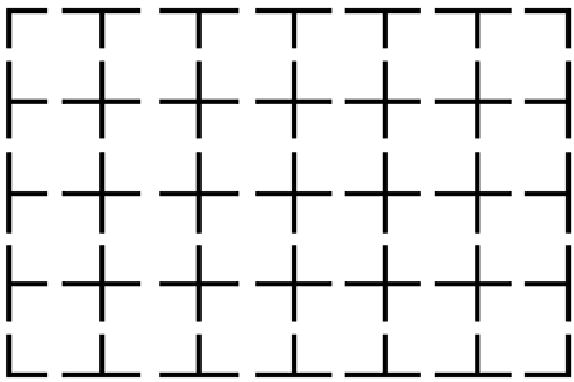


Рис. 2.2.4

Компания по производству пазлов решила освоить принципиально новый тип головоломок. Для этого берется прямоугольная решетка размера $n \times m$, каждый ее столбец и строка разрезаются посередине пополам. После этого образуются фигуры трех типов: четыре уголка, $2 \cdot (n+m-2)$ т-образных фигур и $(n-1) \cdot (m-1)$ крестиков.

Тому, кто решает головоломку, требуется сложить из этих фигур исходную прямоугольную решетку. При этом необходимо использовать абсолютно все имеющиеся в наличии фигуры.

Формат входных данных

В первой строке заданы через пробел два числа a — количество т-образных фигур и b — количество крестиков, которые находятся в одном из пазлов. При этом в наборе всегда есть еще четыре уголка. Известно, что этот комплект позволяет собрать прямоугольную решетку размера $n \times m$, где $1 \leqslant n,\ m \leqslant 10^9$.

Формат выходных данных

Требуется по числам a и b найти размеры исходной решетки n и m. Будем всегда считать, что $n \leqslant m$, то есть нужно вывести в одну строку через пробел два числа, первое из которых не превосходит второго, и вместе они задают размеры загаданной решетки.

Примеры

Пример №1

Стандартный ввод
`16 15`

Стандартный вывод
`4 6`

Пример №2

Стандартный ввод
`0 0`

Стандартный вывод
`1 1`

Комментарий

Задачу можно решить либо бинарным поиском, либо при помощи квадратного уравнения.

Решение

Ниже представлено решение на языке C++ при помощи бинпоиска.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b;
    cin >> a >> b;
    int L = 0, R = a / 4 + 1;
    while(R - L > 1){
        int M = (R + L) / 2;
        int D = a / 2 - M;
        if(M * D <= b){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << L + 1 <<' '<< a / 2 - L + 1 << endl;
}
```

Задача 2.2.3.3. Восемь пирогов и одна свечка (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Мечта Карлсона наконец-то сбылась! Мама Малыша испекла восемь пирогов прямоугольной формы и в один из них воткнула свечку. После того как Карлсон съел семь пирогов, он решил-таки поделиться кусочком оставшегося восьмого пирога с Малышом. Но, будучи в хорошем настроении, он вынул из пирога свечу и предложил ему решить задачку.

«Так как я самый щедрый Карлсон в мире, то делить оставшийся пирог будешь ты. Но учти, ты должен разрезать пирог одним прямым разрезом так, чтобы линия прошла через один из углов и точку, где стояла свечка. После этого я выберу себе один из двух кусочков, а оставшийся, так и быть, достанется тебе».

Малыш не против этого замысла, однако считает, что разрезать пирог нужно как можно более справедливо, то есть так, чтобы разница между меньшим и большим кусками была как можно меньше. Подскажите Малышу, какой минимальной разницы между площадями кусков он сможет добиться.

Формат входных данных

В первой строке находятся два числа n и m через пробел — размеры прямоугольного пирога. Пирог размещен на координатной плоскости так, что его левый нижний угол находится в точке $(0, 0)$, а правый верхний — в точке (n, m), где $2 \leqslant n,\ m \leqslant 1\,000$.

Во второй строке находятся два числа x и y через пробел — координаты свечки, где $1 \leqslant x \leqslant n - 1$, $1 \leqslant y \leqslant m - 1$, то есть свечка находится строго внутри пирога.

Формат выходных данных

Вывести одно вещественное число с точностью не менее трех знаков после десятичной точки — минимальную разницу между площадями двух получающихся после разрезания кусков, которую сможет получить Малыш.

Примеры

Пример №1

Стандартный ввод
8 5 7 2

Стандартный вывод
12.571

Пример №2

Стандартный ввод
2 2 1 1

Стандартный вывод
0.000

Примечания

На рис. 2.2.5 представлены четыре варианта разделения пирога для первого примера из условия. Можно видеть, что самый близкий к справедливому способ разделения связан с разрезом из левого верхнего угла. Площадь треугольника в этом случае будет равна $96 / 7$, площадь четырехугольника равна $184 / 7$, и разница равна $88 / 7$, что при округлении до трех знаков равно 12,571.

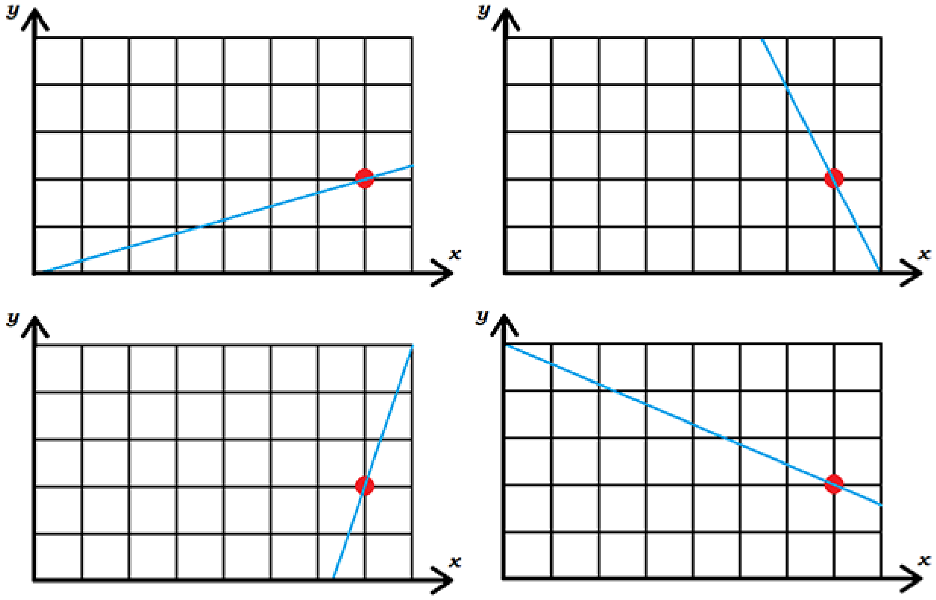


Рис. 2.2.5

Комментарий

Геометрия: для каждого из четырех случаев аккуратно находим катеты прямоугольного треугольника при помощи пропорции, затем находим площадь этого треугольника и, вычитая из всего прямоугольника эту площадь, находим площадь второго куска. Далее выбираем наиболее оптимальное отношение площадей.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
double katy(double x, double y, double n){
    return n * y / x;
}
double n, m, x, y;
double ans = INF;
double k1, k2;
void upd(){
    if(k1 < m){
        double st =k1 * n / 2;
        ans = min(ans, n * m - 2 * st);
    }
    else{
        double st =k2 * m / 2;
        ans = min(ans, n * m - 2 * st);
    }}
signed main(){
    cin >> n >> m >> x >> y;
    k1 = katy(x, y, n);
    k2 = katy(y, x, m);
    upd();
    k1 = katy(n - x, y, n);
    k2 = katy(y, n - x, m);
    upd();
    k1 = katy(x, m - y, n);
    k2 = katy(m - y, x, m);
    upd();
    k1 = katy(n - x, m - y, n);
    k2 = katy(m - y, n - x, m);
    upd();
    cout.precision(3);
    cout << fixed << ans<< endl;
}
```

Задача 2.2.3.4. Плетенка (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

У Маши есть n полосок бумаги. i-я полоска имеет ширину 1 и длину a_i. Маша разделит эти полоски на две части и покрасит некоторые в желтый, а оставшиеся — в зеленый цвет. Она сама выберет, какие полоски как покрасить. Далее она хочет из этих полосок сплести максимально большую плетенку. Она расположит полоски одного цвета в некотором порядке горизонтально, а полоски другого цвета в некотором порядке вертикально. После этого она переплетет горизонтальные и вертикальные полоски так, что они будут чередоваться то сверху, то снизу, образуя в местах пересечения шахматную раскраску. Наконец, она обрежет выступающие края полосок так, что останется прямоугольная плетенка с ровными краями. Каждая клетка полученной плетенки должна иметь два слоя.

Маша хочет сплести максимально большую по площади прямоугольную плетенку. Подскажите ей, плетенку какой площади она сможет сделать. Заметим, что она может при создании плетенки использовать не все имеющиеся у нее полоски.

Формат входных данных

В первой строке на вход подается число n — количество полосок бумаги у Маши, где $2 \leqslant n \leqslant 2 \cdot 10^5$. Во второй строке через пробел заданы n целых чисел a_i через пробел — длины полосок, где $1 \leqslant a_i \leqslant 10^9$.

Формат выходных данных

Вывести одно число — площадь прямоугольника, форму которого может иметь самая большая плетенка Маши.

Примеры

Пример №1

Стандартный ввод
8 3 6 5 4 4 5 5 2

Стандартный вывод
12

Примечания

На рис. 2.2.6 представлен один из вариантов получения самой большой плетенки для полосок из примера. Синим обозначена граница полученной максимальной плетенки. Ее размер 3×4, и ее площадь 12. При ее создании Маша не должна использовать полоску номер 8, по этой причине неважно, как она раскрашена.

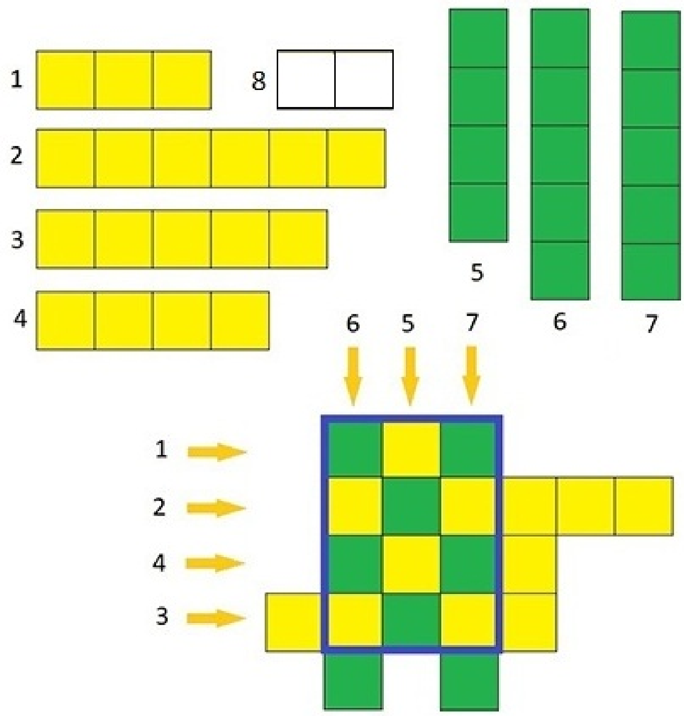


Рис. 2.2.6

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    deque<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    sort(v.begin(), v.end());
    int ans = 0;
    int cnth = 0, minh;
    while(1){
        if(v.size() == 0){
            break;
        }
        cnth++;
        minh = v.back();
        v.pop_back();
        while(v.size() > 0 && v[0] < cnth){
            v.pop_front();
        }
        ans = max(ans, cnth * min(minh, (int)v.size()));
    }
    cout << ans << endl;
}
```

Задача 2.2.3.5. Английский в игровой форме (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

Маша и Витя запоминают слова английского языка в оригинальной игровой форме. За день им нужно выучить n слов, где $20 \leqslant n \leqslant 100$, каждое из которых имеет длину от 5 до 8 символов. Маша выбирает из этого набора наугад несколько попарно различных слов (также от 5 до 8) и собирает их в одну строку без пробелов. Далее она переставляет буквы в этой строке так, что слова оказываются полностью перепутанными, и дает эту строку Вите. Теперь Витя должен восстановить все слова, которые выбрала Маша.

Но у Вити плохо получается, а Маша уже забыла, какие слова она выбрала. Нужно им помочь — написать программу, которая восстановит слова, выбранные Машей.

Формат входных данных

В первой строке находится строка, которую Маша предложила Вите. Во второй строке содержится число n — количество слов, которые нужно выучить детям, $20 \leqslant n \leqslant 100$.

В следующих n строках содержатся эти слова по одному в строке. Все слова в этом наборе различны. Слова отсортированы в лексикографическом (алфавитном) порядке. Все слова состоят из маленьких букв от a до z. Обратите внимание, что в тестах к этой задаче все заданные слова реально существуют в английском языке и случайным образом выбраны из словаря.

Гарантируется, что длина каждого слова из предложенного набора (словаря) в пределах от 5 до 8, строка, которую получила Маша, может быть получена путем перестановки букв некоторых различных слов из предложенного словаря, причем, набор выбранных Машей слов определяется по ней однозначно. Количество слов, из которых составлена Машина строка, находится в пределах от 5 до 8.

Формат выходных данных

Вывести все слова, выбранные Машей, в алфавитном порядке по одному в строке.

Примеры

Пример №1

Стандартный ввод
stirbaexsudueoeidgomttcrnrwlunapntetacwri 24 bridge cranky document drawing farmer fighter figurine gravy havoc minimum reactant reply republic sonata soprano split subset tailor texture tomorrow trout vicinity wrist writer

Стандартный вывод
document drawing republic sonata texture wrist

Комментарий

В случае, выделенном в условии (слова являются случайными, взятыми из английского словаря), задача решается рекурсией с перебором вариантов.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
string frs;
int n;
vector<string> dict;
vector<int> msk(26, 0);
int cnt = 0;
vector<vector<int> > amsk;
vector<string> ans;
bool bigok = 0;
void p(int pos){
    if(!bigok){
        if(cnt == 0){
            sort(ans.begin(), ans.end());
            bigok = 1;
            return;
        }
        for(int i = pos; i < n; i++){
            string ts = dict[i];
            bool ok = 1;
            for(int j = 0; j < 26; j++){
                if(amsk[i][j] > msk[j]){
                    ok = 0;
                }
            }
            if(ok){
                ans.push_back(ts);
                for(int j = 0; j < 26; j++){
                    msk[j] -= amsk[i][j];
                    cnt -= amsk[i][j];
                }
                p(i + 1);
                if(!bigok){
                for(int j = 0; j < 26; j++){
                    msk[j] += amsk[i][j];
                    cnt += amsk[i][j];
                }
                ans.pop_back();
                }
            }
        }
    }
}
signed main(){
    cin >> frs;
    cin >> n;
    amsk.resize(n, vector<int>(26, 0));

    string ts;
    for(int i = 0; i < n; i++){
        cin >> ts;
        dict.push_back(ts);
    }
    for(int i = 0; i < n; i++){
        for(auto el : dict[i]){
            amsk[i][el - 'a']++;
        }
    }
```

```
    for(auto el : frs){
        msk[el - 'a']++;
        cnt++;
    }
    p(0);
    for(auto el : ans){
        cout << el << endl;
    }
}
```

2.2.4. Четвертая волна. Задачи 8–11 класса

Задачи четвертой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63457/enter/.

Задача 2.2.4.1. Квадратный флаг (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одному портному заказали сделать одноцветный флаг. Особенность этого флага в том, что он должен быть квадратным. У портного есть два прямоугольных куска ткани заданного цвета. Один из них имеет размеры $a \times b$, другой — $c \times d$. Так как клиент будет платить пропорционально площади изготовленного флага, портной хочет сначала сшить имеющиеся у него прямоугольные куски, соединив их двумя какими-то сторонами, а затем из полученного полотна вырезать и сделать флаг с максимально большой стороной. Определить сторону получившегося у него флага.

Формат входных данных

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника — целые числа a, b через пробел, во второй — размеры второго прямоугольника, также целые числа c, d через пробел, где $1 \leqslant a, b, c, d \leqslant 10^9$.

Формат выходных данных

Вывести одно число — сторону самого большого квадрата, который можно получить по условию задачи.

Примеры

Пример №1

Стандартный ввод
2 4 3 6

Стандартный вывод
4

Пример №2

Стандартный ввод
2 2 3 6

Стандартный вывод
3

Примечания

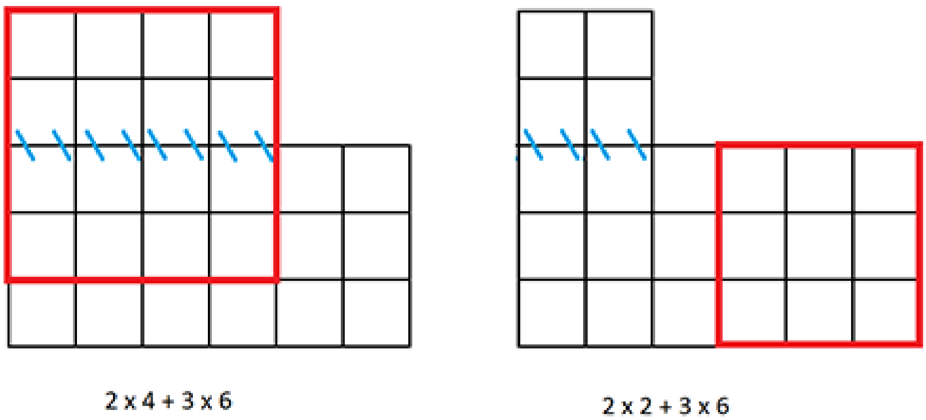


Рис. 2.2.7

На рис. 2.2.7 представлены иллюстрации для тестов из условия. Синими штрихами обозначено место сшивки двух кусков. Красный квадрат выделяет один из вариантов вырезания максимального квадрата.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    int ans = max(min(a, b), min(c, d));
    int p1 = min(a + c, min(b, d));
    int p2 = min(a + d, min(b, c));
    int p3 = min(b + c, min(a, d));
    int p4 = min(b + d, min(a, c));
    ans = max({ans, p1, p2, p3, p4});
    cout << ans << endl;
}
```

Задача 2.2.4.2. Потерянная ДНК (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

В данной задаче будем упрощенно считать, что ДНК представляется строкой длины от 10 до 100, состоящей из букв A, C, G, T.

Пусть даны две ДНК D_1 и D_2 одной и той же длины n. Выберем некоторое произвольное число i от 1 до $n-1$ и поменяем местами префиксы (начала) этих ДНК длины i. Будем говорить, что полученные новые две строки образованы путем скрещивания двух исходных по префиксу длины i.

Например, пусть $D_1 =$ **AACGGTAGGT**, а $D_2 =$ TCCCGGAACA. Выберем $i = 4$ и поменяем местами префиксы длины 4. Получим две новые ДНК, одна из которых будет иметь вид **AACG**GGAACA, а вторая — TCCC**GTAGGT**. Для наглядности были выделены части первой из них.

Полученные новые ДНК снова могут быть скрещены по любому префиксу длины от 1 до $n-1$.

Теперь можно рассмотреть популяцию из нескольких ДНК. Выберем из них две, произведем их скрещивание по префиксу какой-либо длины и поместим две новые ДНК в исходную популяцию. В данной задаче будем считать, что количество ДНК не увеличивается, то есть старые две ДНК заменяются на новые две ДНК.

Дана исходная популяция из m ДНК, каждая имеет одну и ту же длину n. После некоторого количества попарных скрещиваний была получена новая популяция. Но при итоговой обработке данных сведения об одной ДНК из новой популяции были потеряны. Задача состоит в отыскании этой потерянной ДНК по оставшимся $m-1$ ДНК из новой популяции.

Формат входных данных

В первой строке через пробел даны два числа n — длина ДНК и m — количество ДНК в исходной популяции, где $10 \leqslant n \leqslant 100$, $2 \leqslant m \leqslant 100$.

В следующих m строках содержится описание исходной популяции ДНК, каждая задается строкой длины n, состоящей из символов A, C, G и T.

Далее следует разделяющая строка, содержащая n символов «—».

Далее следует еще $m - 1$ строк, описывающих новую (заключительную) популяцию без одной ДНК.

Гарантируется, что данные верны, то есть $m - 1$ последняя ДНК является некоторой новой популяцией ровно без одной ДНК, полученной из исходной популяции, заданной в m первых строках.

Формат выходных данных

Вывести недостающую утерянную ДНК.

Примеры

Пример №1

Стандартный ввод

```
10 2
AACGGTAGGT
TCCCGGAACA
----------
TCCCGTAGGT
```

Стандартный вывод

```
AACGGGAACA
```

Пример №2

Стандартный ввод

```
10 4
AACCGGTTAA
ACGTACGTAC
AAACCCGGGT
CATTACTGGA
----------
AAGCGCTTAA
CCACACGTGC
AACTAGGGGT
```

Стандартный вывод

```
AATTCCTGAA
```

Комментарий

Для каждой позиции нужно найти недостающую букву из первого набора ДНК. Для этого удобнее всего использовать функцию `xor`.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n, m;
    cin >> n >> m;
    vector<string> v1(m);
    for(int i = 0; i < m; i++){
        cin >> v1[i];
    }
    string d;
    cin >> d;
    vector<string> v2(m - 1);
    for(int i = 0; i < m - 1; i++){
        cin >> v2[i];
    }
    for(int j = 0; j < n; j++){
        int ss = 0;
        for(int i = 0; i < m; i++){
            ss ^= (int)(v1[i][j]);
        }
        for(int i = 0; i < m - 1; i++){
            ss ^= (int)(v2[i][j]);
        }
        cout << (char)(ss);
    }
    cout << endl;
}
```

Задача 2.2.4.3. Утомленные туристы (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим следующий вариант известной задачи на перемещение по туннелю группы из четырех человек. В общем виде она выглядит так: четыре туриста хотят пройти по темному туннелю. Имеется один фонарик. По туннелю можно перемещаться либо вдвоем, либо по одному, при этом у тех, кто движется в туннеле, должен быть фонарик в руках. По этой причине движение должно быть следующим: двое переходят туда, один возвращается обратно и приносит фонарик тем, кто еще не перешел. После этого указанный маневр повторяется снова.

У каждого участника своя скорость движения в туннеле. Пусть участники проходят туннель за A, B, C и D мин. Если идут двое, то они движутся со скоростью того, кто идет медленнее. Требуется по заданным временам прохождения туннеля каждого из участников перевести их максимально быстро через туннель.

Немного усложним данную задачу. Введем фактор усталости. А именно, любой участник, пройдя по туннелю, устает и в следующий раз идет уже медленнее. После каждого прохождения туннеля время прохождения любого участника увеличивается на E мин. Например, если участник до начала движения проходит туннель за 1 мин, а показатель усталости E равен 3 мин, то первый раз участник пройдет туннель за 1 мин, второй раз — за 4 мин, третий раз — за 7 мин и т. д.

По заданным A, B, C, D и E узнать, за какое минимальное время можно провести всю группу через туннель согласно указанным правилам.

Формат входных данных

На вход подаются пять чисел. В первой строке через пробел четыре числа A, B, C и D — время прохождения туннеля каждым из четырех участников до того, как они начали движение. Во второй строке содержится число E — величина, на которую увеличивается время прохождения туннеля каждым участником после каждого перемещения. При этом $1 \leqslant A, B, C, D \leqslant 1\,000$, $0 \leqslant E \leqslant 1\,000$.

Формат выходных данных

Вывести одно число — минимальное время прохождения туннеля всей группой.

Примеры

Пример №1

Стандартный ввод
8 9 10 1 3

Стандартный вывод
44

Пример №2

Стандартный ввод
8 9 10 1 0

Стандартный вывод
29

Примечания

В первом примере при прохождении туннеля каждый турист устает и движется медленнее на 3 мин. Покажем, как перевести группу при этом за 44 мин.

Каждую ситуацию будем обозначать следующим образом: слева от двоеточия находятся туристы, которые стоят в начале туннеля, а справа — те, что стоят в конце туннеля. Туриста будем обозначать при помощи числа, соответствующего его текущему времени прохождения туннеля.

Тогда исходная ситуация имеет вид $1, 8, 9, 10 :$.

Сначала идут туристы 1 и 8, каждый после перехода устает на 3 мин, получим ситуацию $9, 10 : 4, 11$, затрачено 8 мин.

Обратно возвращается турист 4, он устает еще на 3 мин. Ситуация становится $7, 9, 10 : 11$, затрачено $8 + 4 = 12$ мин.

Теперь идут туристы 7 и 9, получится ситуация $10 : 10, 11, 12$, затрачено $8 + 4 + 9 = 21$ мин.

Возвращается турист 10, получится $10, 13 : 11, 12$, затрачено $8 + 4 + 9 + 10 = 31$ мин.

Наконец, оставшиеся двое туристов 10 и 13 за 13 мин переходят туннель, итого затрачено $8 + 4 + 9 + 10 + 13 = 44$ мин.

Комментарий

Задача решается рекурсивным перебором всех вариантов прохождения.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
vector<int> v(4);
int e, ans = INF;
void p(vector<int> &vl, vector<int> &vr, int tv){
    if(vl.size() == 2){
        ans = min(ans, tv + *max_element(vl.begin(), vl.end()));
        return;
    }
    for(int i = 0; i < vl.size() - 1; i++){
        for(int j = i + 1; j < vl.size(); j++){
            vector<int> vl1;
            for(int k = 0; k < vl.size(); k++){
                if(k != i && k != j){
                    vl1.push_back(vl[k]);
                }
            }
            vector<int> vr1 = vr;
```

```
                vr1.push_back(vl[i] + e);
                vr1.push_back(vl[j] + e);
                int tmp = max(vl[i], vl[j]);
                sort(vr1.rbegin(), vr1.rend());
                vl1.push_back(vr1.back() + e);
                vr1.pop_back();
                p(vl1, vr1, tv + tmp + vl1.back() - e);
            }
        }
}
signed main(){
    for(int i = 0; i < 4; i++){
        cin >> v[i];
    }
    sort(v.begin(), v.end());
    cin >> e;
    vector<int> vl = v, vr;
    p(vl, vr, 0);
    cout << ans;
}
```

Задача 2.2.4.4. Проектируем мост (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

При постройке моста используются два типа пролетов: П-образные (они прочные, но дорогие) и Т-образные (они дешевле, но менее надежные). Мост должен начинаться и заканчиваться П-образными пролетами. Любой Т-образный пролет должен иметь хотя бы один П-образный пролет в качестве соседнего.

Длина проектируемого моста — n пролетов. Муниципалитет выделил средства на постройку a П-образных и b Т-образных пролетов. При этом $a + b = n$. Требуется выяснить, сколькими способами при этих условиях можно скомпоновать мост. Два способа компоновки моста отличаются, если в одной на некоторой позиции стоит П-образный пролет, а в другой на этой же позиции стоит Т-образный пролет.

Формат входных данных

В одной строке через пробел заданы два числа: a — число П-образных пролетов и b — число Т-образных пролетов, на постройку которых выделены средства, где $2 \leqslant a \leqslant 10^6$, $0 \leqslant b \leqslant 10^6$.

Формат выходных данных

Вывести одно число — количество вариантов компоновки моста. Так как ответ может быть очень большим, требуется вывести остаток от его деления на $1\,000\,000\,007$ $(10^9 + 7)$.

Примеры

Пример №1

Стандартный ввод
`4 3`

Стандартный вывод
`7`

Примечания

Для примера из условия имеется 7 вариантов компоновки моста (пробелы добавлены для лучшего восприятия вариантов):

П Т Т П Т П П

П Т Т П П Т П

П Т П Т Т П П

П Т П П Т Т П

П П Т П Т Т П

П П Т Т П Т П

П Т П Т П Т П

Комментарий

При заданных ограничениях задача решается только при помощи комбинаторики с вычислениями по модулю.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
const int MOD = 1e9 + 7;
vector<int> f(2e6 + 1, 1);
```

```
int binpow (int a, int n) {
        int res = 1;
        while (n > 0) {
                if (n % 2 == 1)
                        (res *= a) %= MOD;
                (a *= a) %= MOD;
                n /= 2;
        }
        return res;
}

int bc(int n, int k){
    int res = f[n];
    int p1 = binpow(f[k], MOD - 2);
    int p2 = binpow(f[n - k], MOD - 2);
    (res *= p1) %= MOD;
    (res *= p2) %= MOD;
    return res;
}
signed main(){
    for(int i = 1; i <= 2e6; i++){
        f[i] = (f[i - 1] * i) % MOD;
    }
    int a, b;
    int ans = 0;
    cin >> a >> b;
    a--;
    for(int i = 0; i < a + 1; i++){
        if(2 * i <= b){
            int d = bc(a, i);
            if(b - 2 * i <= a - i){
                (d *= bc(a - i, b - 2 * i) ) %= MOD;
                (ans += d) %= MOD;
            }
        }
    }
    cout << ans << endl;
}
```

Задача 2.2.4.5. Джентльмены на прогулке (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 8 с.

Ограничение по памяти: 64 Мбайт.

Условие

По прямому участку улицы, которую будем считать отрезком AB длины d, прогуливаются n джентльменов. i-й джентльмен движется со скоростью v_i. Скорости всех джентльменов попарно различны. Дойдя до любого конца улицы, каждый джентльмен поворачивает и идет в обратную сторону.

При каждой встрече два джентльмена приветствуют друг друга, приподнимая головной убор. Приветствие происходит и в том случае, когда один джентльмен обгоняет другого. Если два джентльмена встречаются в момент их одновременного поворота, то происходит два приветствия: одно до поворота, другое — после поворота. Если происходит одновременная встреча трех и более джентльменов, то они приветствуют друг друга попарно, то есть каждый каждого. Допустим, если одновременно встретились четыре джентльмена где-то посреди улицы, произойдет шесть попарных приветствий. Если же эти четыре джентльмена встретились в момент их одновременного поворота, произойдет уже двенадцать приветствий.

В этой задаче считаем, что все действия происходят без остановок, то есть и повороты и приветствия происходят мгновенно. Джентльмены одновременно начинают свою прогулку из точки A в момент 0. В этот момент они уже производят свои первые попарные приветствия, то есть в момент 0 уже произведено $n \cdot (n-1)/2$ приветствий. Момент старта не считается моментом поворота, то есть на старте число приветствий не удваивается. Джентльмены гуляют достаточно долго, чтобы произошло любое заданное количество приветствий.

Требуется найти момент, в который было произведено k-е по порядку приветствие.

Формат входных данных

В первой строке ввода через пробел содержится два целых числа: d — длина отрезка AB и n — количество прогуливающихся джентльменов, где $1 \leqslant d \leqslant 200$, $2 \leqslant n \leqslant 2\,000$.

Во второй строке находятся n целых чисел v_i через пробел — скорости каждого джентльмена, где $1 \leqslant v_i \leqslant 2\,000$. Гарантируется, что все скорости попарно различны. Скорости даны в порядке возрастания, то есть $v_1 < v_2 < \ldots < v_n$.

В третьей строке содержится одно целое число k — номер требуемого приветствия, для которого нужно найти момент, когда оно произойдет, где $1 \leqslant k \leqslant 10^9$.

Формат выходных данных

Вывести одно вещественное число — время, когда произойдет k-е по порядку приветствие. Ответ вывести с точностью не менее двух знаков после десятичной точки.

Примеры

Пример №1

Стандартный ввод
5 4 2 5 8 10 6

Стандартный вывод
0.000

Пример №2

Стандартный ввод
5 4 2 5 8 10 7

Стандартный вывод
0.556

Пример №3

Стандартный ввод
5 4 2 5 8 10 11

Стандартный вывод
1.000

Пример №4

Стандартный ввод
5 4 2 5 8 10 15

Стандартный вывод
1.429

Пример №5

Стандартный ввод
5 4 2 5 8 10 17

Стандартный вывод
1.667

Пример №6

Стандартный ввод
5 4 2 5 8 10 19

Стандартный вывод
1.667

Пример №7

Стандартный ввод
5 4 2 5 8 10 21

Стандартный вывод
2.000

Примечания

На рис. 2.2.8 приведено положение джентльменов из примеров в моменты времени 0, 1 и 2. Джентльмены обозначены своими скоростями. Стрелками обозначены направления их движения в соответствующий момент. Перечислим и пронумеруем в порядке возрастания моменты попарных приветствий этих джентльменов до момента времени 2 включительно. Если два и более приветствия происходят одновременно, неважно какое из них конкретно имеет номер k, главное, что они происходят в один и тот же определенный момент времени.

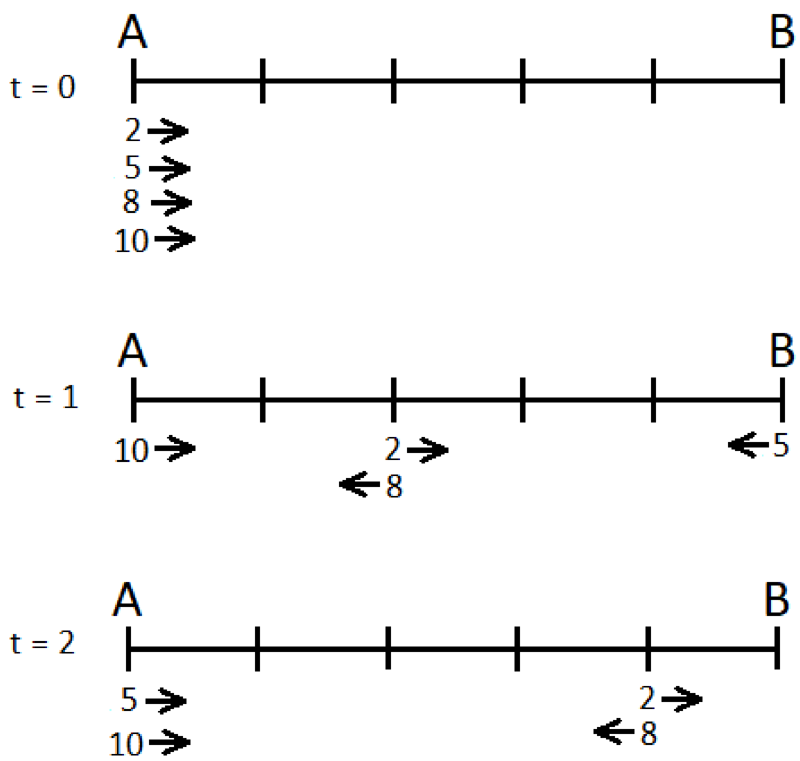


Рис. 2.2.8

1. 2 и 5 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
2. 2 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
3. 2 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
4. 5 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
5. 5 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).

6. 8 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
7. 8 и 10 приветствуют друг друга в момент 0.556.
8. 5 и 10 приветствуют друг друга в момент 0.667.
9. 5 и 8 приветствуют друг друга в момент 0.769.
10. 2 и 10 приветствуют друг друга в момент 0.833.
11. 2 и 8 приветствуют друг друга в момент 1.000 (изображено на рис. 2.2.8).
12. 8 и 10 приветствуют друг друга в момент 1.111.
13. 2 и 10 приветствуют друг друга в момент 1.250.
14. 5 и 10 приветствуют друг друга в момент 1.333.
15. 2 и 5 приветствуют друг друга в момент 1.429.
16. 5 и 8 приветствуют друг друга в момент 1.538.
17. 2 и 8 приветствуют друг друга в момент 1.667.
18. 2 и 10 приветствуют друг друга в момент 1.667.
19. 8 и 10 приветствуют друг друга в момент 1.667 (в момент 1.667 встретятся одновременно три джентльмена 2, 8 и 10).
20. 2 и 8 приветствуют друг друга в момент 2.000 (изображено на рис. 2.2.8).
21. 5 и 10 приветствуют друг друга в момент 2.000 (до поворота).
22. 5 и 10 приветствуют друг друга в момент 2.000 (после поворота, изображено на рис. 2.2.8).

Комментарий

Задача решается при помощи бинпоиска с квадратичным нахождением ответа в каждой его итерации.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const double EPS = 1e-7;
double x(double M, int V, int d){
    double dst = V * M;
    int cnt = floor((dst + EPS) / d);
    double pin = dst - cnt * d;
    if(cnt % 2 == 0){
        return pin;
    }
    else{
        return d - pin;
    }
}
int F(double M, vector<int> &v, int d){
    int res = 0;
    for(int i = 0; i < v.size(); i++){
        double dst = v[i] * M;
```

```
            int cnt = floor((dst + EPS) / d);
            res += cnt * i;
            double tx = x(M, v[i], d);
            for(int j = 0; j < i; j++){
                double txj = x(M, v[j], d);
                if(cnt % 2 == 0){
                    res += txj <= tx + EPS;
                }
                else{
                    res += txj >= tx - EPS;
                }
            }
        }
        return res;
    }
signed main(){
    int d, n;
    cin >> d >> n;
    vector<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    int k;
    cin >> k;
    double L = 0, R = 1;
    while(F(R, v, d) <= k){
        R *= 2;
    }
    R *= 2;
    while(R - L > 1e-4){
        double M = (R + L) / 2.0;
        if(F(M, v, d) < k){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout.precision(10);
    cout << fixed << L << endl;
}
```

2.3. Предметный тур. Математика

2.3.1. Первая волна. Задачи 8–9 класса

Задачи первой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63459/enter/.

Задача 2.3.1.1. (15 баллов)

Тема: арифметика.

Условие

Оля в каждую клетку таблицы 3×3 записала по некоторому числу и с удивлением заметила, что сумма чисел в каждой строке и в каждом столбце таблицы равна 23. Внимательный же одноклассник Витя к ее размышлениям добавил информацию, что сумма чисел в каждом получившемся квадрате 2×2 равна 32. Какое число Оля записала в центральную клетку таблицы?

Решение

Проанализируем исходную таблицу и увидим, что при построении всех возможных квадратов 2×2:

- числа, стоящие в угловых клетках исходной таблицы, входят по одному разу;
- числа, стоящие во второй строке и во втором столбце — по два раза;
- центральное число — четыре раза.

Тогда если найдем сумму чисел во всех квадратах 2×2 и из нее вычтем сумму чисел всей таблицы, а также сумму чисел, стоящих во втором столбце и второй строке, то найдем центральное число, то есть $32 \cdot 4 - 23 \cdot 3 - 23 \cdot 2 = 13$.

Ответ: 13.

Задача 2.3.1.2. (15 баллов)

Тема: комбинаторика.

Условие

Нечетное восьмизначное число назовем «интересным», если оно состоит из простых цифр и одинаковые цифры не стоят рядом. Сколько существует таких «интересных чисел»?

Решение

Простые цифры — это 2, 3, 5 и 7. Тогда так как «интересное» число должно быть нечетным, то в разряде его единиц может стоять только 3, 5 или 7, то есть три варианта. В разряде десятков также может стоять только три варианта, т. к. одинаковые цифры не могут стоять рядом, и т. д. Таким образом, общее количество «интересных» чисел равно $3^8 = 6561$.

Ответ: 6561.

Задача 2.3.1.3. (20 баллов)

Тема: планиметрия.

Условие

В остроугольном треугольнике ABC провели высоты AA_1 и CC_1. Точки E и F — середины отрезков AC и A_1C_1 соответственно.

Найдите длину отрезка EF, если известно, что $AC = 30$ и $A_1C_1 = 24$.

Решение

В прямоугольном треугольнике AC_1C с гипотенузой AC: $C_1E = \dfrac{1}{2}AC = 15$. Аналогично в треугольнике A_1C: $A_1E = \dfrac{1}{2}AC = 15$.

Таким образом, треугольник A_1C_1E является равнобедренным, и его медиана EF является также и высотой.

Тогда по теореме Пифагора: $EF^2 = A_1E^2 - A_1F^2 = 15^2 - 12^2 = 81$, $EF = 9$.

Ответ: 9.

Задача 2.3.1.4. (25 баллов)

Темы: уравнения, формулы сокращенного умножения.

Условие

Найдите значение выражения $x + y + 3z$, если известно, что числа x, y, z удовлетворяют равенству:

$$5x^2 + 4y^2 + 9z^2 + 12z + 13 = 4xy + 12x.$$

Решение

Преобразуем равенство следующим образом:

$$(x^2 - 4xy + 4y^2) + (4x^2 - 12x + 9) + (9z^2 + 12z + 4) = 0,$$

то есть

$$(x-2y)^2+(2x-3)^2+(3z+2)^2=0.$$

Данное равенство будет выполняться при условии, что каждое слагаемое равно 0.

Отсюда получаем систему

$$\begin{cases} x-2y=0, \\ 2x-3=0, \\ 3z+2=0, \end{cases}$$

единственным решением которой будет

$$x=\frac{3}{2};\ y=\frac{3}{4};\ z=-\frac{2}{3}.$$

Тогда

$$x+y+3z=\frac{3}{2}+\frac{3}{4}+3\cdot\left(-\frac{2}{3}\right)=\frac{1}{4}=0{,}25.$$

Ответ: 0,25.

Задача 2.3.1.5. (25 баллов)

Тема: теория вероятностей.

Условие

Шестизначное число будем называть «замечательным», если оно составлено из цифр 1, 2, 3, 4, 5, 6 (каждая цифра используется в числе по одному разу) и кратно 12. Какая вероятность, что сгенерированное компьютером шестизначное число будет «замечательным»?

Ответ выразите в долях и округлите его до четвертого знака после запятой.

Решение

Для того чтобы «замечательное» число делилось на 12, оно должно делиться на три и на четыре. Заметим, что все рассматриваемые числа кратны трем, так как сумма их цифр равна 21.

Для того же чтобы число было кратно четырем, необходимо, чтобы две его последние цифры образовывали число, кратное четырем. В нашем случае это могут быть варианты: 12, 16, 24, 32, 36, 52, 56, 64, всего их восемь. К каждому из них нужно приписать впереди четырехзначное число, составленное из остальных четырех цифр, таких чисел $4!=24$. Значит, всего «интересных» чисел $24\cdot 8=192$.

Всего же шестизначных чисел $9\cdot 10^5=900\,000$.

Тогда вероятность, что сгенерированное компьютером число будет являться «замечательным», будет равна $\dfrac{192}{900\,000}\approx 0{,}0002$.

Ответ: 0,0002.

2.3.2. Первая волна. Задачи 10–11 класса

Задачи первой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63476/enter/.

Задача 2.3.2.1. (10 баллов)

Темы: комбинаторика, десятичная запись числа, цифры.

Условие

Двузначное число назовем подходящим, если оно состоит из четных цифр, расположенных по возрастанию (например, 26). Сколько существует таких подходящих чисел?

Решение

Число не может начинаться с нуля, так что можно использовать только цифры 2, 4, 6, 8. Выпишем все подходящие: 24, 26, 28, 46, 48, 68.

Ответ: 6.

Задача 2.3.2.2. (15 баллов)

Темы: текстовые задачи, пропорции, составление уравнений.

Условие

На Марсе планируется разместить колонию в 100 тысяч человек. Разные колонисты будут заняты на разных работах и важно, чтобы каждый вид работы выполняли группы из минимального количества человек. Одна из важных задач — обеспечение колонистов сбалансированным питанием. Нормы здорового рациона были рассчитаны таким образом, чтобы обеспечить для каждого человека 350 г картофеля в день. Полный цикл производства картофеля от посадки и до сбора составляет 60 дней, каждые 60 дней часть собранного урожая используется для выращивания нового. В той технологии, которую используют космонавты, с 1 га можно вырастить 250 т картофеля, а для посадки нужно 5 т/га. Специальная обработка почвы позволяет добиться сохранения постоянного уровня урожайности, причем можно засадить и обрабатывать произвольную долю гектара. Чтобы полностью обслуживать один гектар в условиях теплиц на Марсе, требуется труд четырех человек.

Какое минимальное количество человек должны трудиться на выращивании картофеля?

Решение

Один человек за 60 дней по плану должен съедать $60 \cdot 0{,}35 = 21$ кг картофеля. Следовательно, 100 тысяч человек по плану за это время съедят 2 100 000 кг.

С одного гектара получаем 250 т, но при этом из них 2 т нужно использовать для посадки. Это значит, что с каждого гектара люди получат в свой рацион 245 т картофеля. Если разделить количество картофеля, которое съест по плану колония за 60 дней, на количество картофеля, которое попадет к ним с 1 га, то получится, что требуется приблизительно 8,571 га. Так как каждый гектар должны обрабатывать четыре человека, то для обработки 8,571 га потребуется труд 34,286 человек. Это значит, что 34 человек недостаточно, требуется запланировать труд 35 человек.

Ответ: 35.

Задача 2.3.2.3. (20 баллов)

Темы: уравнение параболы, координаты вершины параболы.

Условие

Две параболы с различными вершинами пересекаются таким образом, что первая парабола проходит через вершину второй параболы, а вторая — проходит через вершину первой. Уравнение первой параболы имеет вид $y = x^2$, второй $y = a \cdot x^2 + b \cdot x + c$. Найдите, чему равна величина $10 \cdot a + c$.

Решение

Координаты вершины первой параболы имеют вид $(0; 0)$, следовательно, коэффициент $c = 0$. Координаты вершины второй параболы имеют вид

$$x = -\frac{b}{2a};$$
$$y = -\frac{b^2}{4a}.$$

Тогда, подставив их в уравнение первой параболы, получаем:

$$-\frac{b^2}{4a} = \frac{b^2}{2a}.$$

Отсюда $a = -1$.

Ответ: -10.

Задача 2.3.2.4. (25 баллов)

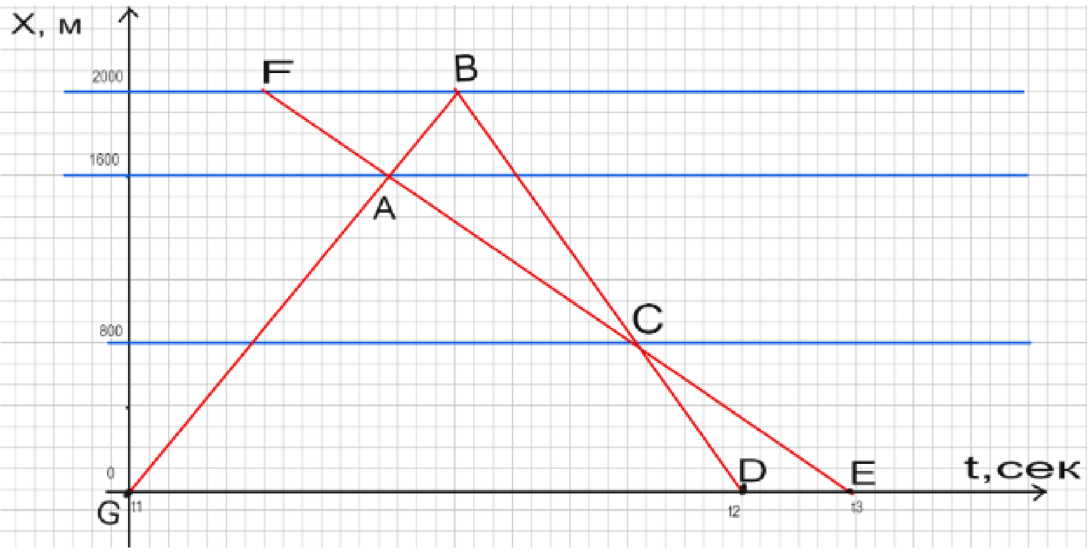
Темы: текстовые задачи и логика, графическое изображение движения, теорема Менелая.

Условие

В 6:00 со дна океана, находящегося на глубине 2 000 м, на поверхность, двигаясь с постоянной скоростью вертикально вверх, начала всплывать подводная лодка. Когда она поднялась до глубины 400 м, капитан заметил, что мимо них вниз плывет глубоководный батискаф. Ему что-то показалось странным. Когда подводная лодка поднялась на поверхность, капитан понял, что на оболочке батискафа были признаки повреждения. Чтобы предотвратить возможную трагедию, в тот же самый момент с подводной лодки вниз спустили спасательный глубоководный аппарат, который спускался с некоторой постоянной скоростью. Когда до дна оставалось 800 м, этот аппарат поравнялся с батискафом. Если бы спасательный аппарат не перехватил батискаф, то спасательный аппарат достиг бы дна к 11:00. Предполагая, что спасательный аппарат все время движения двигался равномерно, определите, в какой момент времени батискаф достиг бы дна, если бы он продолжил движение с той же постоянной скорость. Ответ введите в виде двух целых чисел, записанных подряд — количество часов и количество минут.

Решение

Самое изящное решение получается графическим методом.



Здесь данные из условия задачи можно обозначить следующим образом:

$h_1 = 2\,000 - 400 = 1\,600$, $h(2) = 800$, $H = 2\,000$, $t_1 = 6$, $t_2 = 12$.

По теореме Менелая получаем:

$$\frac{GA}{AB} \cdot \frac{BC}{CD} \cdot \frac{DE}{GE} = 1.$$

Выразим эти отношения из разных пар подобных треугольников:

$$\frac{GA}{AB} = \frac{h_1}{H - h_1};$$
$$\frac{BC}{CD} = \frac{H - h_2}{h_2};$$
$$\frac{DE}{GE} = \frac{t_3 - t_2}{t_3 - t_1};$$
$$\frac{h_1}{H - h_1} \cdot \frac{H - h_2}{h_2} \cdot \frac{t_3 - t_2}{t_3 - t_1} = 1.$$

Подставим числа:

$$\frac{1\,600}{400} \cdot \frac{1\,200}{800} \cdot \frac{t_3 - 11}{t_3 - 6} = 1;$$
$$6 \cdot (t_3 - 11) = t_3 - 6;$$
$$5 \cdot t_3 = 60;$$
$$t_3 = 12 \text{ ч.}$$

Ответ: 12.

Задача 2.3.2.5. (30 баллов)

Условие

Инженер-исследователь работает над созданием новой системы гиперпространственной навигации для космических кораблей, которая потребует меньших вычислительных ресурсов. Часть измерений гиперпространства скрыта от нас и устроена не так, как мы привыкли, а именно — являются дискретными (с конечным количеством позиций), позиции в которых следуют друг за другом циклически. Например, если это измерение, в котором 5 позиций, то их можно занумеровать числами от 0 до 4 так, что космический корабль, при прямолинейном движении вдоль этого измерения, будет пролетать позиции $0 - 1 - 2 - 3 - 4 - 0 - \ldots$ (конечно, корабль в любой момент может изменить направление своего движения на обратное или начать/продолжить изменять позиции и по другим измерениям гиперпространства).

Оказалось, что в гиперпространстве возможна быстрая (но не мгновенная) телепортация: для такого перемещения требуется особая последовательность перемещений в дискретных подпространствах с остановками лишь в выделенные моменты времени. Ранее для хранения таких сложных гипермаршрутов использовалась технология сплошного хранения всех промежуточных опорных точек пути. Однако из-за воздействия агрессивной космической радиации устройства хранения информации часто выходят из строя, что делает сплошное хранение информации очень дорогим, так как требует многократного резервного копирования,

Инженер корабля предложил хранить не сами последовательности позиций, а формулы для их вычисления (что хранить гораздо дешевле и надежнее). В частности, ему удалось запрограммировать движение в одном из измерений с 13 позициями следующим образом: начальное положение обозначается числом 0 и дальнейшие позиции для остановки вычисляются по формуле: x_{n+1} равно остатку от деления $(x_n^5 + 2)$ на 13.

Переход корабля из одной позиции в соседнюю по прямому или обратному ходу занимает 1 единицу времени, которую называют таймом. Корабль, используя эту формулу, прошел полный цикл по остановкам и вернулся в позицию с номером 0.

Какое минимальное количество таймов могло занимать все его движение между остановками в ходе этого цикла?

Решение

Запишем последовательность позиций, в которых останавливается корабль:

$$0 - 2 - 8 - 10 - 6 - 4 - 12 - 1 - 3 - 11 - 9 - 5 - 7 - 0.$$

Между каждыми двумя позициями корабль может двигаться либо прямым ходом, либо обратным. Нужно выбирать кратчайший из двух.

Тогда общая длительность промежутков будет:

$$T = 2 + 6 + 2 + 4 + 2 + 5 + 2 + 2 + 5 + 2 + 4 + 2 + 6 = 44.$$

Ответ: 44.

2.3.3. Вторая волна. Задачи 8–9 класса

Задачи второй волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63460/enter/.

Задача 2.3.3.1. (15 баллов)

Тема: арифметика.

Условие

Первый поезд мимо телеграфного столба проезжает за 9 с, второй поезд мимо этого же столба — за 14 с, а, двигаясь навстречу мимо друг друга, они проезжают за 10 с (с момента, когда поравнялись их начала, и до момента, когда разминулись концы).

Во сколько раз скорость первого поезда больше скорости второго?

Решение

Пусть x м/с — скорость первого поезда, тогда из условия задачи его длина 9 м. Аналогично, если y м/с — скорость второго поезда, то его длина равна $14y$ м.

Зная, что, двигаясь навстречу мимо друг друга, они проезжают за 10 с, составим уравнение:

$$\frac{9x + 14y}{x + y} = 10.$$

Решив это уравнение, получим $x = 4y$. То есть скорость первого поезда в четыре раза больше скорости второго.

Ответ: 4.

Задача 2.3.3.2. (15 баллов)

Тема: *комбинаторика.*

Условие

Вася и Петя играют в разведчиков и для этого придумали свой язык шифрования, в котором используются только пять символов. При этом все «слова» в их сообщениях непустые, то есть содержат хотя бы один знак, и длиной не более пяти знаков.

Сколько различных «слов» они имеют в своем арсенале, чтобы передавать друг другу информацию?

Решение

«Слова», которые могут составлять Вася и Петя на своем языке, могут состоять из 1, 2, 3, 4 и 5 символов.

Тогда общее количество слов будет равно $5^1 + 5^2 + 5^3 + 5^4 + 5^5 = 3905$.

Ответ: 3905.

Задача 2.3.3.3. (20 баллов)

Тема: *геометрия.*

Условие

В треугольнике ABC длина биссектрисы AD равна длине отрезка DC и $AC = 2AB$. Найдите $\angle ABC$.

Решение

В равнобедренном треугольнике ADC из точки D проведем медиану DE на сторону AC, которая также будет являться и высотой.

Тогда $AE = \dfrac{1}{2}AC = AB$. Треугольники AED и ABD равны по двум сторонам и углу между ними: $AE = AB$, AD — общая сторона и $\angle DAE = \angle DAB$.

Следовательно, $\angle ABC = \angle ABD = \angle AED = 90^\circ$.

Ответ: 90°.

Задача 2.3.3.4. (25 баллов)

Тема: десятичная запись натурального числа.

Условие

В натуральном двузначном числе a цифры поменяли местами и получили двузначное число b. Оказалось, что сумма чисел a и b делится на 5, а их разность — на 27.

Найдите все возможные значения числа a. В ответ запишите сумму всех полученных чисел.

Решение

Пусть $a = \overline{xy} = 10x + y$ и $b = \overline{yx} = 10y + x$.

Тогда
$$a + b = 11x + y = 11(x + y).$$

Так как по условию $a + b = 11(x + y) \,\vdots\, 5$ и числа 5 и 11 взаимно просты, то
$$(x + y) \,\vdots\, 5. \tag{2.3.1}$$

Далее из второго условия $a - b = 9x - 9y = 9(x - y) \,\vdots\, 27$, следует, что
$$(x - y) \,\vdots\, 3. \tag{2.3.2}$$

Осталось перебрать все возможные значения цифр x и y, удовлетворяющих условиям (2.3.1) и (2.3.2). Непосредственной проверкой можно убедиться, что этим условиям удовлетворяют пары (1; 4), (2; 8), (4; 1), (5; 5), (6; 9), (8; 2) и (9; 6).

Таким образом, получаем пять чисел, сумма которых равна $14 + 28 + 41 + 55 + 69 + 82 + 96 = 385$.

Ответ: 385.

Задача 2.3.3.5. (25 баллов)

Тема: текстовая задача.

Условие

Команда «Математики» за последние три года, согласно протоколам, приняла участие в 111 матчах по мини-футболу (в это число вошли и игры, которые были отменены по техническим причинам). При анализе результатов было замечено:

- сколько-то игр было выиграно;
- ничьи составляют 45% от всех игр, в которых не были одержаны победы;
- количество матчей, в которых были допущены поражения, к количеству отмененных игр относится как $1:2$.

Какое количество матчей «Математики» проиграли?

Решение

Пусть было одержано x побед. Тогда количество игр, которые были сыграны вничью, проиграны или были отменены, равно $111 - x$.

Тогда $\frac{9}{20}(111 - x)$ — количество игр, сыгранных вничью.

Найдем количество игр, которые были проиграны, или отменены:

$$(111 - x) - \frac{9}{20}(111 - x) = \frac{1221 - 11x}{20}.$$

Тогда количество игр, в которых были поражения, равно

$$y = \frac{1221 - 11x}{60} \in Z.$$

Получили диофантово уравнение

$$11x + 60y = 1221.$$

Выразим x:

$$x = 111 - 60 \cdot \frac{y}{11}.$$

Таким образом, $y \,\vdots\, 11$ и $y > 0$.

Рассмотрим различные случаи относительно y:

1. $y = 11$. Тогда $x = 111 - 60 = 51$.
2. $y = 22$. Тогда $x = 111 - 120 = -9$. Количество игр не может быть отрицательным числом. Следовательно, данный случай, как и все последующие, не подходит.

Таким образом, количество игр, в которых были получены поражения, равно 11.

Ответ: 11.

2.3.4. Вторая волна. Задачи 10–11 класса

Задачи второй волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63477/enter/.

Задача 2.3.4.1. (10 баллов)

Темы: стереометрия, центральная симметрия.

Условие

Прямоугольный параллелепипед имеет объем, равный 30. Его рассекли на две части, проведя плоскость через точку пересечения всех трех его диагоналей.

Чему равно максимальное значение объема одной из этих двух частей?

Решение

При центральной симметрии плоскости переходят в параллельные им плоскости, а прямые — в параллельные им прямые. Диагонали параллелепипеда делятся точкой пересечения пополам, поэтому он имеет центр симметрии. При центральной симметрии любая точка параллелепипеда, не находящаяся на секущей плоскости, перейдет в точку, которая находится с другой стороны от любой плоскости, проходящей через этот центр, так как эти две точки и центр симметрии находятся на одной прямой, которая пересекает эту плоскость.

Таким образом, плоскость делит параллелепипед на две части, которые переходят друг в друга при центральной симметрии. Следовательно, их объемы должны быть равны. Это означает, что часть параллелепипеда имеет объем, равный половине объема параллелепипеда

Ответ: 15.

Задача 2.3.4.2. (15 баллов)

Темы: теорема Виета, многочлены.

Условие

Путешественник достал древнюю карту спрятанных сокровищ на острове Пасхи. Путь к пещере, в которой пиратами был закопан клад, был зашифрован с помощью квадратного уравнения. К сожалению, с течением времени запись одного из коэффициентов стерлась, и поэтому путешественник не смог его точно восстановить.

Оказалось, что оно имеет следующий вид: $x^2 + 6x + a = 0$.

Здесь буквой a обозначен неизвестный коэффициент.

Уравнение использовалось для того, чтобы можно было разделить инструкцию по поиску сокровища на несколько частей таким образом, чтобы совершенно невозможно было бы понять, что и где искать, если хотя бы одной части недостает.

У путешественника были все части инструкции, поэтому он смог понять, что нужно от нужной точки на побережье идти ровно P км на юг вдоль единственной тропы, затем $Q = \dfrac{P}{2}$ км на запад, а потом повернуться на северо-восток и идти прямо, пока вершина вулкана Теревака, кратер Рано-Арои, не станет виден под углом ровно $10R^{\circ}$ над уровнем горизонта. Рядом с этим местом и находится пещера. Здесь P, Q — корни данного квадратного уравнения, упорядоченные по возрастанию, $R = 2P + Q$.

Может ли путешественник, исходя из данных условий, однозначно найти два этих корня?

Если может, напишите в ответ число R. Если не может, напишите в ответ число 0.

Решение

Запишем теорему Виета для квадратного уравнения:

$$\begin{cases} P + Q = -6, \\ PQ = a. \end{cases}$$

В условии указано, что $Q = \dfrac{P}{2}$. Подставив в первое уравнение, получаем, что $P = -4$, $Q = -2$.

Ответ: -10.

Задача 2.3.4.3. (20 баллов)

Темы: арифметическая задача, симметрия.

Условие

Исследователи выращивают экспериментальную культуру грибов. Эти грибы размножаются почкованием. Гриб порождает два новых гриба каждые 4 ч. Только что появившийся гриб слишком маленький, и поэтому он должен еще 6 ч расти, прежде чем размножаться, таким образом, первое потомство от нового гриба возникает лишь через 10 ч после его появления из почки.

Сколько грибов, включая только что появившихся, будет в лаборатории через 28 ч, если изначально там был один гриб, который породит два новых гриба только через 4 ч.

Решение

Самый первый гриб за 28 ч успеет породить только три поколения грибов, так как для появления четвертого поколения нужно 30 ч. Поэтому чтобы ответить на вопрос задачи, нужно посчитать, сколько грибов успеют отпочковаться от грибов, которые породил первый гриб, а потом посчитать также третье поколение.

Первые два гриба, отпочковавшиеся через 4 ч, создадут еще четыре гриба в 14 ч, еще четыре — в 18 ч, еще четыре — в 22 ч и еще четыре в — 26 ч. Всего они породят 16 грибов.

Вторые два гриба, появившиеся через 8 ч, создадут еще четыре гриба в 18 ч, еще четыре — в 22 ч и еще четыре гриба — в 26 ч. Всего они породят 12 грибов.

Третьи два гриба, появившиеся через 12 ч, создадут еще четыре гриба в 22 ч, и еще четыре гриба — в 26 ч. Всего они породят восемь грибов.

Четвертые два гриба, появившиеся через 16 ч, создадут еще четыре гриба в 26 ч.

Пятые два гриба — в 20 ч, шестые два гриба — в 24 ч, а седьмые два гриба в 28 ч не успеют породить никаких новых грибов — это еще шесть грибов.

Таким образом, можно посчитать количество грибов первого и второго поколения:

$$N_1 = 7 \cdot 2 = 14;$$
$$N_2 = 16 + 12 + 8 + 4 = 40.$$

Осталось посчитать третье поколение. Оно образуется в 24 ч и 28 ч из первых четырех грибов из первых двух грибов, в 28 ч из вторых четырех грибов из первых двух грибов и в 28 ч из первых четырех грибов из вторых двух грибов. То есть еще восемь грибов:

$$N_3 = 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 = 32.$$

Суммарно получаем:

$$N = 1 + N_1 + N_2 + N_3 = 1 + 14 + 40 + 32 = 87.$$

Ответ: 87.

Задача 2.3.4.4. (25 баллов)

Темы: прямоугольный треугольник, теорема Пифагора, теорема косинусов.

Условие

В прямоугольном равнобедренном треугольнике ABC с прямым углом C проведена биссектриса AL. Из точки L к стороне BC проведен перпендикуляр, который пересек сторону AB в точке M. Перпендикуляр, построенный к стороне AB в точке M, пересекает сторону AC в точке N.

Чему равен угол ANL? Ответ приведите в градусах.

Решение

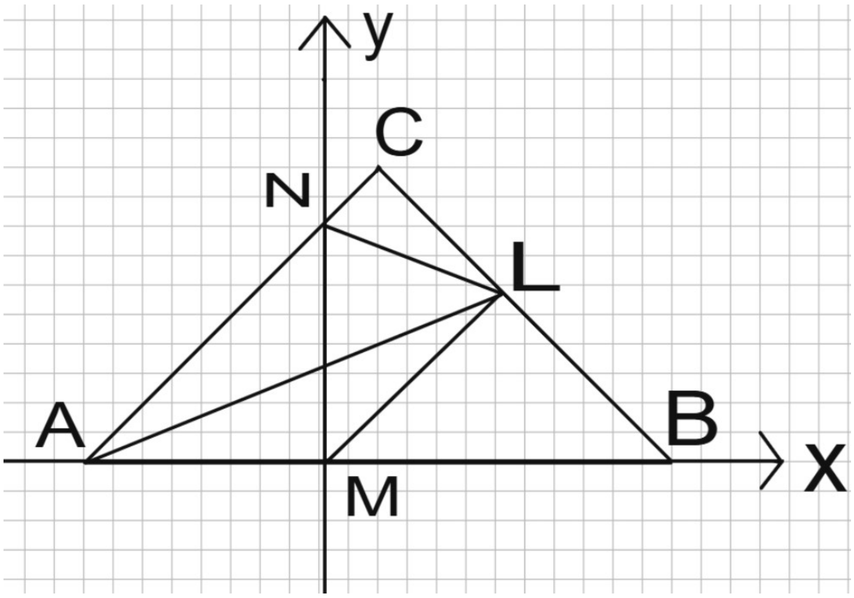
$MN = AM$, значит, угол ANM тоже равен 45° и NM перпендикулярно AB.

Тогда углы NML и BML тоже равны по 45°.

Пусть $AM = 1$ (в условии никаких длин нет, поэтому можем за единицу длины взять любой отрезок). Тогда

$$AN = \sqrt{2};$$

$$AL = 2 \cdot AM \cdot \cos 22{,}5^\circ = 2 \cdot \sqrt{\frac{\sqrt{2}+2}{4}} = \sqrt{\sqrt{2}+2}.$$



Чтобы найти NL, используем метод координат. Проведем горизонтальную ось через AB, вертикальную ось через MN. Тогда точка N имеет координаты (0; 1). Что же касается точки L, то ее координаты x и y совпадают, а длина ML равна 1. Следовательно, они равны $L\left(\frac{\sqrt{2}}{2}; \frac{\sqrt{2}}{2}\right)$.

Используя формулу расстояния между двумя точками, получаем:

$$NL^2 = \frac{1}{2} + \left(1 - \frac{\sqrt{2}}{2}\right)^2 = 2 - \sqrt{2}.$$

Обозначив угол ALN за x, применим теорему косинусов:

$$2 = 2 - \sqrt{2} + \sqrt{2} + 2 - 2 \cdot \sqrt{\sqrt{2}+2} \cdot \sqrt{2-\sqrt{2}} \cdot \cos x.$$

Отсюда получаем, что:

$$\cos x = \frac{\sqrt{2}}{2},\ x = 45^\circ.$$

Тогда угол ANL равен 180° минус угол ALN и угол NAL:

$$ANL = 180 - 45 - 22{,}5 = 112{,}5.$$

Ответ: 112,5.

Задача 2.3.4.5. (30 баллов)

Темы: уравнение параболы, уравнение касательной, угловой коэффициент наклона прямой.

Условие

Для разработки оптической системы на основе параболических отражателей света потребовалось исследовать оптические свойства парабол. Пусть парабола задана уравнением $y = 16x^2$. Требуется на плоскости найти такую точку O, что все проекции этой точки на касательные к параболе лежат на оси абсцисс. Найдите координаты точки O и запишите их в ответ.

Уравнение касательной прямой к параболе (в заданной точке (x_0, y_0)) однозначно устанавливается как уравнение невертикальной прямой, проходящей через (x_0, y_0) и имеющей единственную точку пересечения с параболой.

Решение

Рассмотрим точку с абсциссой x_0 на параболе. Уравнение прямой, проходящей через эту точку, в общем виде имеет вид:

$$y = a \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Приравняем его к уравнению параболы и найдем, при каком значении a они будут иметь ровно одну точку пересечения:

$$16 \cdot x^2 = a \cdot (x - x_0) + 16 \cdot (x_0)^2;$$

$$16 \cdot x^2 - a \cdot x + x_0 \cdot a - 16 \cdot (x_0)^2 = 0;$$

$$D = a^2 - 4 \cdot 16 \cdot (x_0 \cdot a - 16 \cdot x_0^2) = (a - 32 \cdot x_0)^2 = 0;$$

$$a = 32 \cdot x_0.$$

Итак, запишем уравнение касательной в этой точке к параболе в виде

$$y = 32 \cdot x_0 \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Эта прямая пересечет ось абсцисс в точке с координатой $x_1 = \dfrac{x_0}{2}$.

Уравнение прямой, проходящей через эту точку перпендикулярно касательной:

$$y = -\frac{2 \cdot x - x_0}{64 \cdot x_0}.$$

Эта прямая пересечет ось ординат в точке с координатами $(0; 0{,}015625)$. Координаты этой точки не зависят от значения x_0, а значит, все такие прямые пройдут через эту точку.

Ответ: $(0; 0{,}015625)$.

2.3.5. Третья волна. Задачи 8–9 класса

Задачи третьей волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63461/enter/.

Задача 2.3.5.1. (15 баллов)

Тема: *текстовая задача.*

Условие

Начинающий предприниматель Петров закупил 1 000 единиц некоторого товара и попытался его продать с наценкой 20% за единицу продукции. Однако ожидания предпринимателя не совпали с реальностью, и он смог продать только 40% от своего объема, после чего вынужден был снизить цену на товар на 10%. В результате снижения единица товара стала стоить 5 832 руб. за штуку.

Какую чистую прибыль, то есть разность между деньгами, полученными за продажу товара и затратами на его закупку, получил Петров?

Решение

Пусть x руб. — цена за единицу товара, по которой совершена закупка предпринимателем Петровым. Тогда он первоначально планировал осуществить продажи по цене

$$x + 0{,}2x = 1{,}2x.$$

После снижения же цены товар стал стоить

$$1{,}2x - 0{,}1 \cdot 1{,}2x = 1{,}08x.$$

Так как известно, что после снижения единица товара стала стоить 5 832 руб. за штуку, то

$$1{,}08x = 5\,832x = 5\,400.$$

Таким образом, товар был закуплен 5 400 руб. за штуку, и общие затраты на его покупку составили 5 400 000 руб.

Согласно условию задачи 400 единиц товара было продано по цене $1{,}2 \cdot 5\,400 = 6\,480$ руб., и всего было получено за них $6\,480 \cdot 400 = 2\,592\,000$ руб.

Оставшиеся же 600 единиц были проданы по цене 5 832 руб. и получено за них $5\,832 \cdot 600 = 3\,499\,200$ руб.

Тогда чистая прибыль предпринимателя Петрова будет равна

$$2\,592\,000 + 3\,499\,200 - 5\,400\,000 = 691\,200 \text{ руб.}$$

Ответ: 691 200.

Задача 2.3.5.2. (15 баллов)

Тема: *комбинаторика.*

Условие

Сколько существует нечетных пятизначных чисел, в которых есть хотя бы одна цифра 5?

Решение

Для того чтобы найти количество требуемых чисел, достаточно из общего количества пятизначных нечетных чисел вычесть количество чисел, в которых отсутствует цифра 5.

В десятичной записи нечетного пятизначного числа на последнюю позицию претендует пять вариантов (цифры 1, 3, 5, 7 и 9), на первую — девять вариантов (все цифры, кроме нуля), а на все остальные позиции — по 10 вариантов. Тогда общее количество пятизначных нечетных чисел будет равно

$$9 \cdot 10 \cdot 10 \cdot 10 \cdot 5 = 45\,000.$$

Для записи нечетного пятизначного числа, в десятичной записи которого отсутствует цифра 5, на каждую соответствующую позицию будет на один вариант меньше, тогда общее количество таких чисел будет равно

$$8 \cdot 9 \cdot 9 \cdot 9 \cdot 4 = 23\,328.$$

Тогда количество пятизначных нечетных чисел, в которых присутствует хотя бы одна цифра 5, равно

$$45\,000 - 23\,328 = 21\,672.$$

Ответ: 21 672.

Задача 2.3.5.3. (20 баллов)

Темы: алгебра, система уравнений.

Условие

Наблюдательный Витя для некоторых двух различных чисел заметил интересную особенность: первое число, увеличенное на 4, будет равно квадрату второго числа, уменьшенного на 2; и наоборот, если ко второму числу прибавить 4, то результат будет равен квадрату первого числа, уменьшенного на 2. Найдите сумму квадратов данных двух чисел.

Решение

Пусть x, y — два исходных различных числа. Тогда согласно условиям задачи будем иметь систему уравнений:

$$\begin{cases} x + 4 = (y - 2)^2, \\ y + 4 = (x - 2)^2. \end{cases}$$

Вычитая из первого равенства второе, получим:

$$x - y = (y - 2)^2 - (x - 2)^2 = (y - x)(x + y - 4).$$

Так как числа x, y различны, то отсюда получаем, что $x+y=3$.

Складывая же уравнения полученной системы, получим

$$x+y+8=(y-2)^2+(x-2)^2=y^2-4y+4+x^2-4x+4.$$

Из последнего равенства получаем, что

$$x^2+y^2=5(x+y)=15.$$

Ответ: 15.

Задача 2.3.5.4. (25 баллов)

Темы: *теория чисел, остатки.*

Условие

Петя записал на доске три числа 391, 604, 888 и задумчиво сказал Васе: «Если я сейчас эти три числа разделю на одно и то же натуральное число, отличное от единицы, то в результате получу один и тот же остаток».

На какое натуральное число Петя планирует произвести деление исходных чисел?

Решение

Обозначим число, на которое производится деление, через x, а остаток через y.

Тогда каждое из записанных Петей чисел можно представить в виде:

$$\begin{aligned} 391&=xm+y,\\ 604&=xk+y,\\ 888&=xn+y, \end{aligned}$$

где m, k и n — неполные частные, возникающие при делении.

Вычитая из третьего равенства второе, а из второго — первое, получим:

$$\begin{aligned} 284&=x(n-k),\\ 213&=x(k-m). \end{aligned}$$

Вычтем из верхнего равенства нижнее:

$$71=x(n-2k+m).$$

Так как 71 — это простое число, то $71=71\cdot 1$, и, по условию задачи $x\neq 1$, то единственный возможный вариант для делителя Пети равен 71.

Ответ: 71.

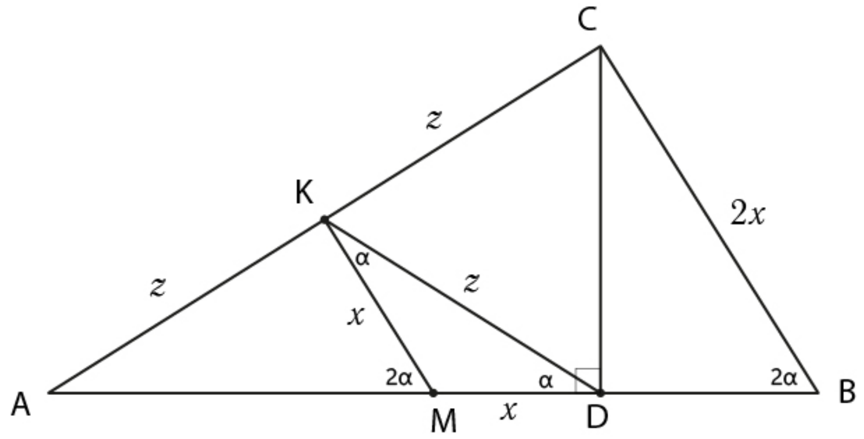
Задача 2.3.5.5. (25 баллов)

Тема: планиметрия.

Условие

CD — высота остроугольного треугольника ABC, M — середина стороны AB и $\angle ABC = 2\angle BAC$. Найдите отношение $BC : MD$.

Решение



На стороне AC отметим ее середину — точку K.

Тогда $AK = KC$ и $AM = MB$ (по условию задачи), следовательно, MK — средняя линия треугольника ABC и $BC = 2MK$.

Докажем, что $MK = MD$.

По свойству медианы прямоугольного треугольника, проведенной из вершины прямого угла, в треугольнике ADC: $DK = \frac{1}{2}AC = AK$.

Таким образом, треугольник AKD — равнобедренный и $\angle KAD = \angle KDA$ как углы при основании KD.

Так как MK — средняя линия треугольника ABC, то $MK \parallel BC$ и $\angle AMK = \angle ABC = 2\angle BAC = 2\angle KAD = 2\angle KDA = 2\angle KDM$.

По теореме о внешнем угле для треугольника MKD

$$\angle AMK = \angle KDM + \angle MKD.$$

Тогда из последних двух равенств следует, что $\angle KDM = \angle MKD$ и треугольник MKD — равнобедренный.

Следовательно, $MK = MD$, и так как $BC = 2MK = 2MD$, то $BC : MD = 2 : 1$.

Ответ: 2.

2.3.6. Третья волна. Задачи 10–11 класса

Задачи третьей волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63478/enter/.

Задача 2.3.6.1. (10 баллов)

Темы: осевая симметрия, равнобедренный треугольник, движение.

Условие

Известно, что выпуклая фигура Ф на плоскости устроена таким образом, что она симметрична относительно любой прямой, которая проходит через точку O на этой плоскости. Самое большое расстояние между двумя точками, принадлежащими фигуре Ф, равно дроби, в числителе которой шесть, а в знаменателе квадратный корень из числа π.

Чему равна площадь фигуры Ф?

Решение

1. Докажем, что все точки на границе фигуры равноудалены от центра симметрии O.
   - Возьмем на границе фигуры произвольную точку A. Пусть расстояние $OA = R$.
   - Возьмем любую другую точку B на границе фигуры.
   - Построим прямую I, проходящую через точку O и являющуюся биссектрисой угла AOB.
   - По свойству осевой симметрии, точка A', симметричная точке A относительно прямой I, также принадлежит фигуре Ф.
   - Поскольку I — биссектриса, точка A' попадет на луч OB. Так как при симметрии расстояние до центра сохраняется ($OA' = OA = R$), точка A' совпадет с точкой B только если $OB = R$.
   - Предположим, что $OB > R$. Тогда точка A лежит внутри отрезка OB. Но поскольку фигура выпуклая, весь отрезок OB должен принадлежать фигуре, а значит, и точка A не может быть граничной. Пришли к противоречию.
   - Предположим, что $OB < R$. Тогда точка B лежит внутри отрезка OA. Это также противоречит тому, что B — граничная точка.
   - Следовательно, единственно возможный вариант — $OB = R$.
   - Поскольку точка B была выбрана на границе произвольно, получается, что все точки границы фигуры Ф находятся на одинаковом расстоянии R от точки O. По определению, это окружность.
2. Так как границей является окружность, то сама фигура — круг.
3. Используя данное в условии значение диаметра ($6/\sqrt{\pi}$), находим радиус ($3/\sqrt{\pi}$) и вычисляем площадь, которая равна 9.

Ответ: 9.

Задача 2.3.6.2. (15 баллов)

Темы: *составление уравнений, составление пропорций, проценты.*

Условие

Находясь на борту космического корабля, главный двигатель за первый час израсходовал 40% всего запаса анобтаниума, а вспомогательные двигатели вместе за это же время израсходовали лишь 300 г анобтаниума. За следующий час главный двигатель израсходовал 80% оставшегося топлива, а вспомогательные двигатели израсходовали 100 г топлива на двоих. В итоге на борту корабля осталось 800 г топлива. Сколько килограммов фантастического топлива было на борту до начала полета?

Решение

Найдем массу анобтаниума, оставшегося к концу первого часа.

Не было израсходовано главным двигателем к этому моменту $100 + 800 = 900$ г.

Это составляет $100 - 80 = 20\%$.

Составим пропорцию и решим ее:

$$\begin{aligned} 20\% &- 900, \\ 100\% &- ? \end{aligned}$$

Значит, к концу первого часа оставалось $900 : 0{,}2 = 4\,500$ г анобтаниума.

Найдем массу топлива к началу первого часа.

Не было израсходовано главным двигателем к этому моменту $4\,500 + 300 = 4\,800$ г, что составляет $100 - 40 = 60\%$.

Составим пропорцию и решим ее:

$$\begin{aligned} 60\% &- 4\,800, \\ 100\% &- ? \end{aligned}$$

Значит, к началу первого часа было $4\,800 : 0{,}6 = 8\,000$ г, что составляет 8 кг.

Ответ: 8.

Задача 2.3.6.3. (20 баллов)

Темы: *уравнение параболы, уравнение прямой.*

Условие

Известно, что три различные точки $A(2;4)$, $B(x;6)$, $C(6;y)$ расположены на координатной плоскости таким образом, что через них нельзя провести параболу с вертикальной осью. При этом также известно, что x — минимальное натуральное подходящее число, неравное единице.

Найдите величину $x + y$.

Решение

Через три точки нельзя провести параболу тогда и только тогда, когда они расположены на одной прямой. Действительно, прямая не может пересекать параболу в трех точках, так как квадратное уравнение имеет не больше двух корней. С другой стороны, если три точки не лежат на одной прямой, то через них всегда можно провести параболу. Покажем это.

Пусть на числовой прямой есть три точки с координатами (x_1, y_1), (x_2, y_2), (x_3, y_3). Запишем уравнение параболы в следующем виде:

$$y = y_1 \frac{(x - x_2) \cdot (x - x_3)}{(x_1 - x_2) \cdot (x_1 - x_3)} + y_2 \frac{(x - x_1) \cdot (x - x_3)}{(x_2 - x_1) \cdot (x_2 - x_3)} + y_3 \frac{(x - x_1) \cdot (x - x_2)}{(x_3 - x_1) \cdot (x_3 - x_2)}.$$

Первое слагаемое равно нулю во второй и третьей точке, и равно y_1 в первой, аналогичным образом устроены второе и третье слагаемые, так что это уравнение задает функцию, проходящую через три точки. Однако надо еще проверить, что уравнение задает именно параболу. Для этого нужно, чтобы коэффициент при x_2 не равнялся нулю.

$$\frac{y_1}{(x_1 - x_2) \cdot (x_1 - x_3)} + \frac{y_2}{(x_2 - x_1) \cdot (x_2 - x_3)} + \frac{y_3}{(x_3 - x_1) \cdot (x_3 - x_2)} \neq 0;$$
$$y_1 \cdot (x_2 - x_3) - y_2 \cdot (x_1 - x_3) + y_3 \cdot (x_1 - x_2) \neq 0.$$

Можно убедиться, что это условие означает, что три точки не лежат на одной прямой. А именно, нахождение трех точек на одной прямой можно записать следующим образом:

$$\frac{y_1 - y_2}{x_1 - x_2} = \frac{y_1 - y_3}{x_1 - x_3};$$
$$(y_1 - y_2) \cdot (x_1 - x_3) = (y_1 - y_3) \cdot (x_1 - x_2);$$
$$y_1 \cdot (x_1 - x_3) - y_1 \cdot (x_1 - x_2) = y_2 \cdot (x_1 - x_3) - y_3 \cdot (x_1 - x_2);$$
$$y_1 \cdot (x_2 - x_3) + y_3 \cdot (x_1 - x_2) - y_2 \cdot (x_1 - x_3) = 0.$$

Таким образом, если это условие выполнено, то через три точки проходит прямая, и не проходит никакая парабола. А если оно не выполнено, то проходит единственная парабола, и нельзя провести никакую прямую.

Тогда выразим угловой коэффициент этой прямой тремя разными способами:

$$k = \frac{2}{x - 2} = \frac{y - 6}{6 - x} = \frac{y - 4}{4}.$$

Отсюда получаем, что

$$y = \frac{4 \cdot x}{x - 2}.$$

Минимальное натуральное x, не равное единице, которое подходит — это $x = 3$. Тогда $y = 12$.

Значит, $x + y = 15$.

Ответ: 15.

Задача 2.3.6.4. (25 баллов)

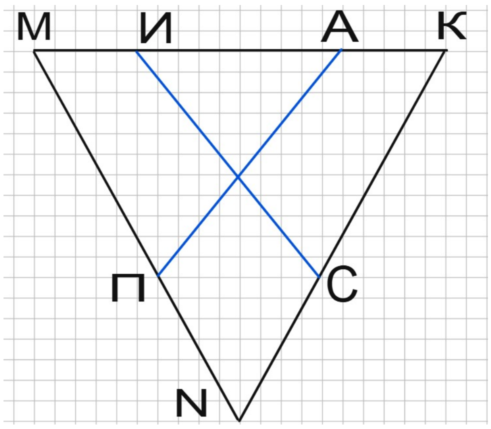
Темы: равносторонний треугольник, первый признак равенства треугольников.

Условие

Три прямые дороги образуют треугольник с равными сторонами, длина которых равна 1 000 м. У этих дорог стоят четыре человека, каждый на обочине одной из трех дорог. Иван и Александр стоят возле одной дороги в 500 м друг от друга. Сергей и Петр стоят у обочин двух других дорог. Сергею идти до Ивана 1 500 м по дорогам кратчайшим путем, Александру до Петра тоже. Между дорогами расположено поле. Какая величина получится, если к расстоянию от Сергея до Ивана по прямой (то есть по полю, а не по дорогам) добавить половину расстояния от Ивана до Александра вдоль дороги, возле которой они стоят, и вычесть расстояние от Александра до Петра по прямой (по полю)?

Решение

Нарисуем расположение всех этих четырех человек. Расположение Сергея и Петра здесь определяется из того условия, что путь до Ивана и Александра соответственно должен занимать 1 500 м, в то время как расстояние по одной стороне не больше 1 000 м, а по другой не больше 500 м.



Используя указанные расстояния, можем записать:

МА + МП = КИ + КС, а значит, МИ + МП = КА + КС = 1 000 м.

МИ + КА = ИА = 500 м. Кроме того, длина КМ равна 1 000 м.

Отсюда выходит, что КА = 1 000 − КС = 1000 − МА, а значит, КС = МА. Аналогично выходит, что КИ = МП.

Тогда треугольники КИС и МПА равны друг другу по двум сторонам и углу между ними.

Следовательно, $АП = СИ$, а значит, $АП - СИ + 0{,}5 \cdot ИА = 250$ м.

Ответ: 250.

Задача 2.3.6.5. (30 баллов)

Темы: делители числа, произведение делителей, разложение на множители.

Условие

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей (рассматриваются все делители, включая само число и единицу). Третья часть всех делителей не делится на 3. Половина четных делителей делится на 5. Само число при этом не превосходит 10 000. Напишите в ответ максимальное число, которое подходит под этим условия.

Решение

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей.

Это значит, что оно делится на 2^5 степени, но не делится на 2^6.

Третья часть всех делителей не делится на 3.

Это значит, что оно делится на 3^2, но не делится на 3^3.

Половина четных делителей делится на 5.

Это значит, что оно делится на 5, но не делится на 25.

Если перемножим 2^5 на 3^2 и на 5, то получим 1440. Минимальное число, подходящее под условия выше, но большее этого числа, равно $7 \cdot 1\,440 = 10\,080 > 10\,000$.

Следовательно, под все условия подходит только число 1 440.

Ответ: 1 440.

2.3.7. Четвертая волна. Задачи 8–9 класса

Задачи четвертой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63462/enter/.

Задача 2.3.7.1. (15 баллов)

Темы: теория чисел, признаки делимости.

Условие

На доске записано число 202420252026. Танечка хочет убрать несколько цифр из исходного числа так, чтобы получившийся результат делился на 45 и являлся наибольшим из всех возможных. Какое число запишет на доске Танечка?

Решение

Для того чтобы число Танечки было бы кратно 45, необходимо выполнение условий делимости на 5 и 9. Следовательно, число должно заканчиваться на 0 или на 5. В данном случае первым делом Танечка должна убрать последние две цифры и получить 2024202520.

Для выполнения условия делимости на 9 необходимо, чтобы сумма цифр числа была бы 9. Сумма цифр сейчас равна 19. Ближайшая сумма, кратная 9, равна 18, но 1 в числе нет, следовательно, следующий вариант — 9. Для этого из оставшегося числа ей нужно вычеркнуть цифры, дающие в сумме 10. Тогда наибольшее число, которое может получить Танечка, — 202050.

Ответ: 202050.

Задача 2.3.7.2. (15 баллов)

Тема: десятичная запись натурального числа.

Условие

Найдите все трехзначные натуральные числа $\overline{abc}$, удовлетворяющие условию

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca}.$$

В ответ запишите сумму всех найденных чисел.

Решение

Распишем равенство, заданное в условии задач

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca};$$
$$100a + 10b + c = 10a + b + 10b + c + 10c + a;$$
$$100a + 10b + c = 11a + 11b + 11c;$$
$$89a = 10c + b.$$

Так как a, b, c — цифры, то единственным решением данного уравнения является набор $a = 1$, $b = 9$, $c = 8$. Следовательно, единственное число, удовлетворяющее условию задачи, это 198.

Ответ: 198.

Задача 2.3.7.3. (20 баллов)

Темы: алгебра, квадратный трехчлен.

Условие

Найдите количество значений параметра b, при которых все корни уравнения $x^2 + bx + 2026 = 0$ целые.

Решение

Пусть x_1 и x_2 — целые корни данного уравнения. Тогда согласно теореме Виета:

$$x_1 \cdot x_2 = 2026.$$

Так как 2026 раскладывается на множители

$$2026 = 1 \cdot 2026 = 2 \cdot 1013,$$

то получаем четыре набора для значений корней

$$(1; 2026),\ (-1; -2026),\ (2; 1013),\ (-2; -1013).$$

Зная значения корней, также по теореме Виета найдем значения параметра b:

$$b = -(x_1 + x_2).$$

Таким образом, всего существует четыре значения параметра $b = \{-2027;\ 2027;\ -2015;\ 2015\}$, при каждом из которых уравнение имеет целые корни.

Ответ: 4.

Задача 2.3.7.4. (25 баллов)

Тема: геометрическая вероятность.

Условие

В треугольнике ABC на биссектрисе BD отмечена точка E так, что $BE = ED$. Найти вероятность, что точка, брошенная в треугольник ABC, попадет в треугольник AED, если $AB = 3$ и $BC = 5$.

Ответ выразите в долях и при необходимости округлите его до четвертого знака после запятой.

Решение

Согласно определению геометрической вероятности, требуемая вероятность будет равна отношению площадей треугольников AED и ABC. AE — медиана треугольника ABE, следовательно, $S_{ABD} = 2S_{AED}$.

Площади треугольников ABD и BDC относятся как длины их оснований AD и DC, то есть

$$\frac{S_{ABD}}{S_{BDC}} = \frac{AD}{DC} = \frac{AB}{BC} = \frac{3}{5}.$$

Последнее равенство выполняется согласно свойству биссектрисы BD в треугольнике ABC. Тогда

$$S_{ABC} = S_{ABD} + S_{BDC} = S_{ABD} + \frac{5}{3}S_{ABD} = \frac{8}{3}S_{ABD} = \frac{16}{3}S_{AED}.$$

Из последнего равенства следует отношение

$$\frac{S_{AED}}{S_{ABC}} = \frac{3}{16} = 0{,}1875.$$

Таким образом, вероятность того, что точка брошенная в треугольник ABC, попадет в треугольник AED, равна 0,1875.

Ответ: 0,1875.

Задача 2.3.7.5. (25 баллов)

Тема: алгебра.

Условие

При каком значении числа a сумма квадратов чисел x и y будет принимать наибольшее значение, если известно, что сумма этих чисел равна $2a+1$, а произведение равно $4a^2 + 8a - 4$?

Решение

По условию задачи $x + y = 2a + 1$ и $xy = 4a^2 + 8a - 4$.

Воспользуемся формулой квадрата суммы двух чисел

$$(x + y)^2 = x^2 + 2xy + y^2.$$

Отсюда

$$x^2 + y^2 = (x + y)^2 - 2xy = (2a + 1)^2 - 2(4a^2 + 8a - 4) = 4a^2 + 4a + 1 - 8a^2 - 16a + 8 =$$

$$= -4a^2 - 12a + 9 = -(4a^2 + 12a + 9) + 18 = -(2a + 3)^2 + 18.$$

В полученном выражении первое слагаемое принимает неположительные значения при любом a. Следовательно, сумма квадратов чисел x и y будет максимальной при $2a + 3 = 0$ или $a = -1{,}5$.

Проверим, что при данном значении параметрам $a = -1{,}5$ числа x и y действительно существуют. В этом случае $x + y = -2$ и $xy = -7$.

Выразив из первого равенства $y = -x - 2$ и подставив его во второе, после преобразований получим уравнение $x^2 + 2x - 7 = 0$. Дискриминант данного уравнения равен 32, следовательно, корни уравнения существуют, по которым однозначным образом восстанавливаются решения построенной системы. Откуда и следует существования чисел x и y, заданных в условии задачи.

Ответ: $-1{,}5$.

2.3.8. Четвертая волна. Задачи 10–11 класса

Задачи четвертой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63479/enter/.

Задача 2.3.8.1. (10 баллов)

Темы: кратчайший путь, параллельный перенос.

Условие

Склад находится в месте, отмеченном на карте точкой A. Нужно проложить дорогу до берега реки, затем построить мост, перпендикулярный течению реки, и от другого берега проложить дорогу до деревни, отмеченной на карте точкой B. Пример подобного построения на рисунке.

Берега реки здесь нарисованы как параллельные прямые. Координатная ось Ox на рисунке отсчитывает положения моста относительно реки в километрах. В примере, приведенном на рисунке, мост проходит через метку 1 км.

Через какую метку должен проходить мост, чтобы сумма длин пути от склада A до реки по дороге и от противоположного берега реки до деревни B была наименьшей? Ответ дайте в километрах.

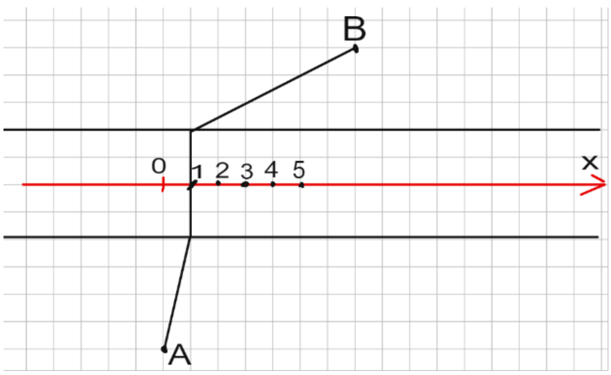


Рис. 2.3.1

Решение

Если вырезать с карты реку и соединить точки A и B прямой, то это и будет кратчайший путь, их соединяющий. Чтобы получить путь до реки, нужно после этого вновь вставить реку. Продемонстрируем эти операции с помощью рис. 2.3.2–2.3.3.

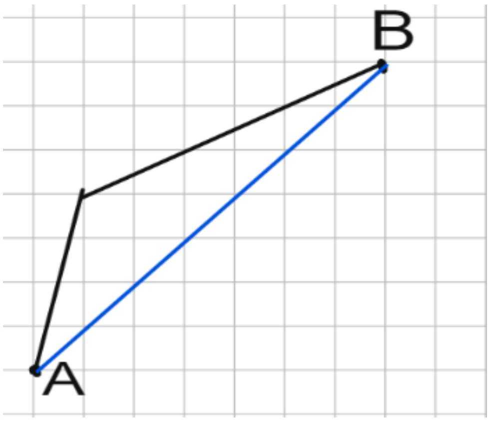


Рис. 2.3.2

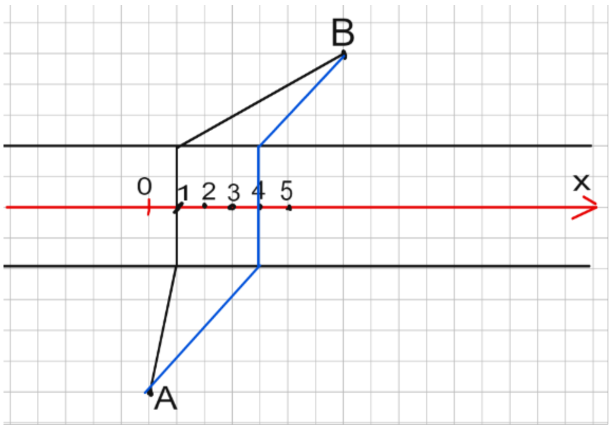


Рис. 2.3.3

Ответ: 4.

Задача 2.3.8.2. (15 баллов)

Темы: составление уравнений, составление пропорций, решение уравнений.

Условие

Два космических корабля стартуют одновременно с одной планеты и направляются к Альфе Центавра, расстояние до которой составляет 4,37 св. лет. Один корабль движется со скоростью 0,1 св. год в год, а другой — со скоростью 0,2 св. год в год.

Через сколько лет расстояние до Альфы Центавра для более быстрого корабля будет в три раза меньше, чем для более медленного корабля? Ответ приведите с точностью до сотых.

Решение

Обозначим время, прошедшее с начала пути, как t лет.

Расстояние, пройденное медленным кораблем, равно $0{,}1t$ св. год, и оставшееся расстояние до Альфы Центавра для медленного корабля $4{,}37 - 0{,}1t$ св. год.

Расстояние, пройденное быстрым кораблем, равно $0{,}2t$ св. год, и оставшееся расстояние до Альфы Центавра для быстрого корабля $4{,}37 - 0{,}2t$ св. год.

По условию задачи, остаток пути для быстрого корабля в 3 раза меньше, чем остаток пути для медленного корабля:

$$4{,}37 - 0{,}2t = \frac{1}{3}(4{,}37 - 0{,}1t).$$

Умножим обе стороны на 3:

$$3(4{,}37 - 0{,}2t) = 4{,}37 - 0{,}1t;$$

$$13{,}11 - 0{,}6t = 4{,}37 - 0{,}1t.$$

Переносим все t в одну сторону и постоянные в другую:

$$13{,}11 - 4{,}37 = 0{,}6t - 0{,}1t;$$

$$8{,}74 = 0{,}5t.$$

Делим обе стороны на 0,5:

$$t = \frac{8{,}74}{0{,}5} = 17{,}48.$$

Таким образом, искомое время равно 17,48 лет.

Ответ: 17,48.

Задача 2.3.8.3. (20 баллов)

Темы: квадратный трехчлен, функции, неопределенные коэффициенты.

Условие

Функция $f(x)$ является квадратным трехчленом и может быть описана следующим образом:

$$f(x) = (f(1) + f(-1) + f(0)) \cdot x^2 + (f(1) + 2 \cdot f(0)) \cdot x - 1.$$

В то же время квадратный трехчлен в общем виде может быть записан так:

$$f(x) = a \cdot x^2 + b \cdot x + c.$$

Найдите минимальное значение величины $a^2 + 2b^2 + 3c^2$ при данных условиях.

Решение

Подставим $f(x)$ в общем виде в первую формулу из условия:

$$a \cdot x^2 + b \cdot x + c = (a + b + c + a - b + c + c) \cdot x^2 + (a + b + c + 2 \cdot c) \cdot x - 1;$$

$$\begin{cases} a = 2 \cdot a + 3 \cdot c, \\ b = a + b + 3 \cdot c, \\ c = -1. \end{cases}$$

Отсюда получаем, что $a = 3$, $c = -1$. Чтобы искомая величина была минимальной, нужно, чтобы коэффициент $b = 0$.

Ответ: 12.

Задача 2.3.8.4. (25 баллов)

Темы: делители числа, произведение делителей, разложение на множители.

Условие

Произведение всех делителей числа 1 000, включая само это число и единицу, равно 10^k.

Чему равно k?

Решение

$$1\,000 = 2^3 \cdot 5^3.$$

Комбинируя все возможные способы выбрать степень двойки и степень пятерки, входящие в делитель, получаем все $(3+1)\cdot(3+1) = 16$ вариантов, каждый из которых соответствует делителю числа. При этом эти 16 делителей можно разбить на пары, произведение в каждой дает 1 000:

$$1\,000 = 1 \cdot 1\,000 = 2 \cdot 500 = 4 \cdot 250 = 8 \cdot 125 = 5 \cdot 200 = 10 \cdot 100 = 20 \cdot 50 = 25 \cdot 40.$$

Тогда выходит, что это будет число $1\,000^8$, а значит, 10^{24}.

Ответ: 24.

Задача 2.3.8.5. (30 баллов)

Темы: равносторонний треугольник, первый признак равенства треугольников.

Условие

Дан квадрат $ABCD$. На сторонах CB и CD отмечены точки L и K соответственно такие, что $CL = CK$. Из точки C на отрезок LD опущен перпендикуляр в точку E.

Пусть $AE = 60$, $EK = 91$. Найдите длину AK.

Решение

Сделаем рис.2.3.4.

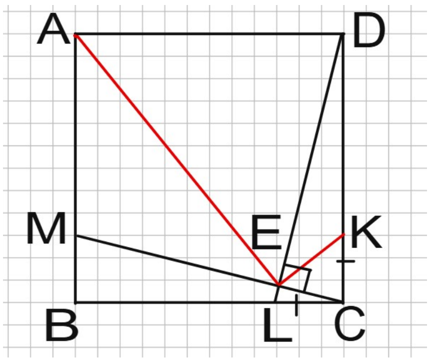


Рис. 2.3.4

Одно из возможных решений заключается в использовании метода координат. Обозначим длину квадрата за единицу, а $CL = LK = x$.

Тогда

$$L(1-x;0);\ K(1;x);\ \overrightarrow{LD} = (x;1);\ \overrightarrow{CE} = t\cdot(1;-x);\ E(1+t;-x\cdot t);\ \overrightarrow{LE} = (t+x;-x\cdot t).$$

Так как вектора LE и LD должны быть сонаправлены, то

$$\frac{t+x}{x} = -x\cdot t;\ t = -\frac{x}{1+x^2};\ E(1-\frac{x}{1+x^2};\ 1-\frac{1}{1+x^2});$$

$$\overrightarrow{AE} = (1-\frac{x}{1+x^2};\ -\frac{1}{1+x^2});\ \overrightarrow{EK} = (\frac{x}{1+x^2};\ x-\frac{x^2}{1+x^2}).$$

Посчитаем скалярное произведение: $\overrightarrow{AE}\cdot\overrightarrow{EK} = 0$. Это значит, что треугольник AEK прямоугольный.

Тогда AK можно найти по теореме Пифагора: $60^2 + 91^2 = 109^2$.

Ответ: 109.

2.4. Инженерный тур

Задача имеет целью проверить навыки работы участников с алгоритмами классификации для создания модели, способной автоматически классифицировать группу животных на фото с фотоловушек. Здесь проверяются их компетенции в обработке данных и применении алгоритмов машинного обучения, а также способность погружаться в предметную сферу и применять алгоритмы компьютерного зрения в решении.

Задача помогает оценить участников по их знаниям и навыкам в области программирования, зоологии, обработки изображений, машинного обучения и применении технических решений для реальных проблем.

Задача 2.4.1. Многоклассовая классификация животных по фото с фотоловушек (100 баллов)

Темы: программирование, исследование данных, машинное обучение, компьютерное зрение.

Условие

Задача посвящена классификации животных на укрупненные группы. Для этого участники получают доступ к уникальным данным — изображениям с фотоловушек, предоставленным исследовательскими организациями-партнерами. Решение задачи позволит снизить нагрузку на ученых, связанную с распознаванием видов, а следовательно, сделать мониторинг популяций более эффективным.

Почему это важно?

Мониторинг популяции диких животных — важная задача для мирового сообщества исследователей дикой природы. Ее решение помогает узнать, какие виды животных находятся под угрозой исчезновения, как они себя ведут в разные периоды жизни, где обитают и многое другое.

Для мониторинга животных используют различные инструменты, в том числе и фотоловушки — специальные камеры, устанавливаемые в лесу и реагирующие на движение в кадре. Каждый год с этих камер приходят сотни тысяч фотографий, на которых нужно найти и категоризировать животных. Это очень сложная и кропотливая работа, так как среди видов, общее число которых может достигать нескольких сотен, могут встречаться и такие, которые визуально друг от друга трудно различимы. К тому же в кадр в момент активации камеры может попасть лишь часть животного, и из-за того, что активность многих видов приходится на ночное время суток, некоторые фотографии могут быть смазанными или засвеченными из специфики работы камер в ночное время.

Описание данных

Датасет представляет собой набор изображений и файл `train.csv` с соответствием каждого изображения определенной группе животных: https://disk.yandex.ru/d/MiCI_OW7U1r_Dg/dataset.

Ниже представлена связь между индексом группы и ее наименованием:

0 — заяц,
1 — кабан,
2 — кошки,
3 — куньи,
4 — медведь,
5 — оленевые,
6 — пантеры,
7 — полорогие,
8 — собачие,
9 — сурок.

Обратите внимание, что в датасете присутствуют сложные примеры: фото, когда в кадр попала небольшая часть животного, а также ночные и смазанные снимки.

Разметка по группам производилась специалистами-исследователями, занимающимися мониторингом состояния популяции живой природы.

Данные разделены на тренировочную и тестовую выборки в соотношении примерно 70% / 30%.

Тестовая часть разбита на публичную и приватную в соотношении примерно 40% / 60%.

Критерии оценивания

Это задача многоклассовой классификации, и в качестве метрики была выбрана `macro F1 score` (подсчет `F1` по каждому классу с дальнейшим усреднением).

В файле представлен фрагмент кода для расчета метрики и оценки алгоритмов участников: https://disk.yandex.ru/d/MiCI_OW7U1r_Dg/ai-stage1.py.

Решение

Требуется разработать алгоритм для классификации группы животных по фото с фотоловушек и представить его результат работы в виде CSV-файла с колонками `image_name`, `predicted_class`.

Ссылка на эталонное экспертное решение: https://disk.yandex.ru/d/MiCI_OW7U1r_Dg/baseline.ipynb.

Ответ.

Решением задачи будет CSV-файл с предсказанием, загружаемый на платформу проведения соревнования для оценки качества решения.

Пример загружаемого файла с решением: `https://disk.yandex.ru/d/MiCI_OW7U1r_Dg/sample_submission.csv`.

А также решение, которое заняло самые высокие позиции в приватном лидерборде: `https://disk.yandex.ru/d/MiCI_OW7U1r_Dg/best_submission.csv`.

3. Второй отборочный этап

3.1. Работа наставника НТО на этапе

На втором отборочном этапе НТО участникам предстоит решать как индивидуальные, так и командные задачи в рамках выбранного профиля. Подготовка к этому этапу требует от них не только глубокого понимания предметной области, но и умения работать в команде, эффективно распределять роли и применять полученные знания на практике. Наставник играет здесь важную роль — он помогает участникам выстроить осмысленную и целенаправленную траекторию подготовки.

Вот основные направления, в которых наставник может поддержать участника:

- **Подготовка по образовательным программам НТО.** Наставник может готовить участников, используя готовые образовательные программы по технологическим направлениям, рекомендованные организаторами, а также адаптировать их под уровень подготовки школьников.
- **Разбор заданий прошлых лет.** Изучение задач второго отборочного этапа прошлых лет помогает участникам понять формат заданий, определить типовые ошибки и выработать стратегии решения.
- **Онлайн-курсы.** Участники могут пройти курсы по разбору задач прошлых лет или курсы, рекомендованные разработчиками отдельных профилей. Наставник может включить эти курсы в план подготовки, а также сопровождать процесс изучения и помогать с возникшими вопросами.
- **Анализ материалов профиля.** Совместный разбор методических материалов, размещенных на страницах профилей, помогает уточнить требования к участникам и направить подготовку на ключевые темы.
- **Практикумы.** Это важный элемент подготовки, позволяющий применять знания на практике. Наставник может:
  - ◇ организовать практикумы по методическим материалам с сайта профиля;
  - ◇ декомпозировать задачи заключительного этапа прошлых лет на отдельные элементы и проработать их с участниками;
  - ◇ провести анализ требуемых профессиональных компетенций и спланировать занятия для развития наиболее значимых из них;
  - ◇ направить участников на практикумы и мероприятия от организаторов, которые анонсируются в официальных сообществах НТО, например, в телеграм-канале для наставников: https://t.me/kruzhok_association.
- **Командная работа.** Одной из ключевых задач наставника на втором этапе является помощь в формировании команды или в поиске подходящей. Наставник может помочь участникам определить их сильные стороны, выбрать роль в команде и сориентироваться в процессе командообразования, включая участие в бирже команд в рамках конкретного профиля.

Если участники не прошли отборочный этап

Случается, что несмотря на усилия и серьезную подготовку, участники не проходят во второй или заключительный этап Олимпиады. В такой ситуации особенно важна поддержка наставника.

- **Поддержка и признание усилий.** Наставнику важно подчеркнуть ценность пройденного пути: полученные знания, навыки, преодоленные трудности и личностный рост. Это помогает участникам сохранить мотивацию и не воспринимать результат как окончательное поражение.
- **Рефлексия.** Полезно организовать встречу для обсуждения впечатления от участия, трудности, с которыми столкнулись школьники и то, что они узнали о себе и команде. Наставник может направить разговор в конструктивное русло: какие выводы можно сделать? Что сработало хорошо? Что можно улучшить?
- **Анализ ошибок и пробелов.** Наставник вместе с участниками анализирует, какие темы вызвали наибольшие затруднения, чего не хватило в подготовке — теоретических знаний, практических навыков, командного взаимодействия. Это позволяет выстроить более эффективную стратегию на будущее.
- **Планирование дальнейшего пути.** Участникам можно предложить:
    - ◇ продолжить углубленное изучение профиля или смежных направлений;
    - ◇ заняться проектной деятельностью, которая укрепит знания и навыки;
    - ◇ сформировать план по подготовке к следующему циклу НТО, начиная с работы над типовыми заданиями и курсами.
- **Создание устойчивой мотивации.** Важно показать школьникам, что участие в НТО — это не просто соревнование, а часть большого образовательного маршрута. Даже неудачный результат может стать толчком к профессиональному росту, если воспринимать его как точку развития, а не как конец пути.

Таким образом, наставник помогает участникам не только готовиться к этапам НТО, но и справляться с неудачами, выстраивать долгосрочную стратегию и сохранять интерес к инженерному и технологическому творчеству.

3.2. Инженерный тур

Задача имеет целью выявить навыки работы участников с алгоритмами классификации для разработки модели, способной автоматически классифицировать группу животных на фото с фотоловушек. В процессе выполнения заданий проверяются компетенции участников в обработке данных и применении алгоритмов машинного обучения, а также способность погружаться в предметную сферу и использовать алгоритмы компьютерного зрения в решении задач.

Задача помогает оценить участников по их знаниям и навыкам в области программирования, зоологии, обработки изображений, машинного обучения и применении технических решений для решения реальных проблем.

3.2.1. Индивидуальная задача

Задача посвящена многоклассовой детекции животных.

Для этого участники получат доступ к уникальным данным — изображениям с фотоловушек, предоставленным исследовательскими организациями-партнерами. Решение задачи позволит отфильтровать пустые изображения, изображения с людьми, транспортом и выделить изображения с животными по группам.

Все материалы и данные к задаче доступны по ссылке: https://disk.yandex.ru/d/hbZfV-2Dq9qJVQ.

Задача 3.2.1.1. Многоклассовая детекция животных по фото с фотоловушек (100 баллов)

Темы: нейросети, информатика.

Условие

Описание задачи

Мониторинг популяции диких животных — важная задача для мирового сообщества исследователей дикой природы. Ее решение помогает узнать, какие виды животных находятся под угрозой исчезновения, как они себя ведут в разные периоды жизни, где обитают и многое другое.

Для мониторинга животных используют различные инструменты, в том числе и фотоловушки — специальные камеры, устанавливаемые в лесу и реагирующие на движение в кадре. Каждый год с этих камер приходят сотни тысяч фотографий, на которых нужно найти и категоризировать животных. Это очень сложная и кропотливая работа, так как общее число видов может достигать нескольких сотен, могут встречаться и такие, которые визуально друг от друга трудно различимы. К тому же в кадр в момент активации камеры может попасть лишь часть животного, и из-за того, что активность многих видов приходится на ночное время суток, некоторые фотографии могут быть смазанными или засвеченными из специфики работы камер в ночное время.

В рамках Олимпиады НТО участникам предлагается помочь ученым автоматизировать рутинную работу по обработке данных с фотоловушек, обучив для этого модели машинного обучения.

Для первичной фильтрации сырых данных и дальнейшей обработки (аналитики и изучения каждой отдельной особи) потребуется разработать многоклассовый детектор фотографий животных на отдельные группы, организованные по принципу схожести видов, имеющихся в данных (например, сибирская косуля, пятнистый олень, изюбрь и марал отнесены к оленевым).

Метрика

Для оценки качества работы модели многоклассовой детекции используется метрика mAP 0,5–0,95 — подсчет средней точности (AP — Average Precision) для каждого класса при различных значениях порога IoU (Intersection over Union) от 0,5 до 0,95 с шагом 0,05 (минимальное IoU для рассмотрения положительного совпадения) с дальнейшим усреднением по классам.

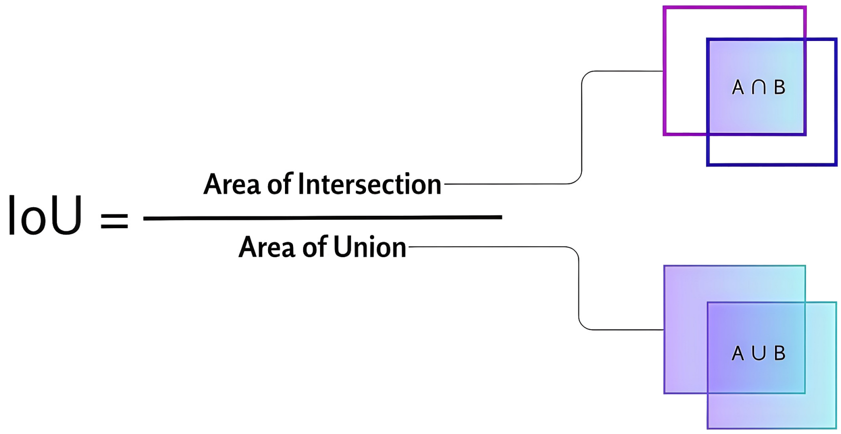


Рис. 3.2.1

Для каждого класса c вычисляется среднее значение AP по всем порогам IoU:

$$\mathrm{mAP}_c = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_{c,\tau_i},$$

где $N = 10$ — количество значений порога IoU (от 0,5 до 0,95 с шагом 0,05).

Среднее значение AP по всем классам (mAP) рассчитывается как:

$$\mathrm{mAP} = \frac{1}{C}\sum_{j=1}^{C} \mathrm{mAP}_j,$$

где C — общее количество классов.

Значение метрики варьируется от 0 (в худшем случае) до 1, если все ограничивающие прямоугольники точно совпадают с истинными и правильно предсказаны классы для них.

Подсчет метрики производится автоматически на платформе при отправке решения.

Данные

Данные доступны по ссылке: https://disk.yandex.ru/d/hbZfV-2Dq9qJVQ.

Датасет представляет собой набор изображений и файл `train.csv` с соответствием каждого изображения определенной группе животных и координатами ограничивающих прямоугольников для каждого животного на изображении.

Координатами ограничивающих прямоугольников является центр ограничивающего прямоугольника и его размеры (ширина и высота) в нормализованном виде в промежутке от 0 до 1. Ширина и высота изменяется в диапазоне от 0 до 1.

Рис. 3.2.2

Ниже представлена связь между индексом группы и ее наименованием:

0 — заяц,
1 — кабан,
2 — кошки,
3 — куньи,
4 — медведь,
5 — оленевые,
6 — пантеры,
7 — полорогие,
8 — собачие,
9 — сурок.

Обратите внимание, что в датасете присутствуют сложные примеры: фото, когда в кадр попала небольшая часть животного, много животных перекрывающих друг друга, а также ночные и смазанные снимки.

Данные были размечены специалистами по разметке данных совместно с учеными зоологами.

Данные разделены на тренировочную и тестовую выборки в соотношении примерно 70% / 30%.

Тестовая часть разбита на публичную и приватную в соотношении примерно 40% / 60%.

Baseline (базовое решение)

baseline_detection.ipynb — `jupyter-notebook`, позволяющий пройти путь от установки библиотек и обучения модели до получения файла с предсказаниями и посмотреть визуально результаты работы алгоритма.

Формат входных данных

Решения принимаются в виде zip-архива размером не более 5 Gb, имеющего структуру:

- `metadata.json`;
- `entry_point` — python-файл;
- файл с весами модели;
- любые дополнительные файлы, необходимые для работы решения (python-файлы, директории, модели и их веса);
- `submission.zip` — zip-архив, содержащий пример решения для загрузки на платформу и получения метрики.

Текущей директорией для запуска решения и под загрузки весов модели будет являться корень архива.

`metadata.json` имеет следующий вид:

json
```json
{
   "image": "odsai/nto24-baseline:1.0",
    "entry_point": "python -u run.py"
}
```

- `image` — поле с названием docker-образа на `https://hub.docker.com/`, в котором будет запускаться решение и содержать все необходимые библиотеки;
- `entry_point` — команда, при помощи которой запускается файл с предсказанием run.py.

Этот файл должен иметь обязательные параметры:

- `--img_dir` — путь до папки с тестовыми изображениями на платформе;
- `--output_path` — путь, по которому проверочная система ожидает результат решения.

Алгоритм для получения предсказания от модели должен сформировать файл с предсказанием, на основании которого происходит расчет метрики. Сам файл представляет собой csv-документ с двумя колонками image_name и predicted_detection. Строки — это пары с названием файла изображения и соответствующего ему список предсказаний.

Список предсказаний модели имеет следующий формат:

$$label_1\ cx_1\ cy_1\ w_1\ h_1\ confidence_1; label_2\ cx_2\ cy_2\ w_2\ h_2\ confidence_2;\ \dots\ ;$$
$$label_n\ cx_n\ cy_n\ w_n\ h_n\ confidence_n,$$

где

- $label_i$ — идентификатор группы, к которому относится животное;
- (cx_i, cy_i) — координата центра i-го прямоугольника (в относительных значениях);
- w_i, h_i — ширина и высота i-го прямоугольника (в относительных значениях);
- $confidence_i$ — уверенность модели детекции в предсказании от 0 до 1.

В случае отсутствия предсказанного ограничивающего прямоугольника сформировать предсказание в виде пустой строки ("").

Пример формирования корректного файла с предсказаниями.

Структура файла с предсказаниями, см. таблицу 3.2.1.

Таблица 3.2.1

image_name	predicted_detection
cc27b9b56583a615fb3.JPG	0 0.5 0.5 0.25 0.25 0.9
087872711fe672676fd.JPG	0 0.5 0.5 0.5 0.5 0.9;0 0.5 0.5 0.5 0.5 0.9
424aa1aa8eb5bbdd07.JPG	-
c5537eaa60525efd7b.JPG	0 0.5 0.5 0.25 0.25 0.9
e9f15b67ca49453e28.JPG	0 0.5 0.5 0.25 0.25 0.9

Онлайн-курс по работе с docker: https://ai-academy.ru/training/courses/docker-git/.

Ограничения по памяти и ресурсам

В течение одного дня участник может загрузить для оценки не более пяти решений. Учитываются только валидные попытки, получившие численную оценку.

Контейнер с решением запускается в следующих условиях:

- 96 GB оперативной памяти;
- Nvidia v100 32 GB;
- время на выполнение решения: 10 мин на public, 10 мин на private;
- решение не имеет доступа к ресурсам интернета;
- максимальный размер упакованного архива с решением: 5 GB.

Использование сторонних датасетов

Допускается использование открытых (доступных в сети) датасетов с лицензией, позволяющей свободное некоммерческое использование.

Решение

Требуется разработать алгоритм для детекции групп животных по фото с фотоловушек и представить его результат работы в виде `csv`-файла с колонками `image_name`, `predicted_detection`.

Решением задачи является `csv` файл с предсказанием, пример файла: `sample_submission.zip`.

Разбор эталонного решения доступен по ссылке: https://gist.github.com/ntomaterials/0e543eb9c086ce419211d5c57ecbe272.

3.2.2. Командная задача

Задача посвящена классификации животных.

Для этого участники получают доступ к уникальным данным — изображениям с фотоловушек, предоставленным исследовательскими организациями-партнерами. Решение задачи должно распределять все изображения с животными на заданные виды животных и организовать оперативный мониторинг раненных животных, контроль выхода диких животных в населенный пункт.

Все материалы к задаче доступны по ссылке: https://disk.yandex.ru/d/Jx8fZJTSsk_LeQ.

Задача 3.2.2.1. Многоклассовая классификация животных по фото с фотоловушек (100 баллов)

Темы: нейросети, информатика.

Условие

Описание задачи

Мониторинг популяции диких животных — важная задача для мирового сообщества исследователей дикой природы. Ее решение помогает узнать, какие виды животных находятся под угрозой исчезновения, как они себя ведут в разные периоды жизни, где обитают и многое другое.

Для мониторинга животных используют различные инструменты, в том числе и фотоловушки — специальные камеры, устанавливаемые в лесу и реагирующие на движение в кадре. Каждый год с этих камер приходят сотни тысяч фотографий, на которых нужно найти и категоризировать животных. Это очень сложная и кропотливая работа, так как общее число видов может достигать нескольких сотен, могут встречаться и такие, которые визуально друг от друга трудно различимы. К тому же в кадр в момент активации камеры может попасть лишь часть животного, и из-за того, что активность многих видов приходится на ночное время суток, некоторые фотографии могут быть смазанными или засвеченными из специфики работы камер в ночное время.

В рамках Олимпиады НТО участникам предлагается помочь ученым автоматизировать рутинную работу по обработке данных с фотоловушек, обучив для этого модели машинного обучения.

Для мониторинга численности каждого вида животного участникам необходимо разработать классификатор фотографий животных по видам.

Метрика

Для оценки качества работы модели используется macro F1 score: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html` (pdf-версия страницы `https://disk.yandex.ru/i/tRtB4WGtGfSF1Q`).

Его особенность заключается в вычислении среднего арифметического F1 score по каждому классу. Значение метрики варьируется от 0 (в худшем случае) до 1, если все предсказания были корректны.

Подсчет метрики производится автоматически на платформе при отправке решения.

Описание данных

Датасет представляет собой набор изображений и файл `train.csv` с соответствием каждого изображения определенному виду животных.

Ниже представлена связь между индексом вида и его наименованием:

- 0 — бурый медведь,
- 1 — гималайский медведь,
- 2 — кабан,
- 3 — изюбрь,
- 4 — пятнистый олень,
- 5 — марал,
- 6 — сибирская косуля,
- 7 — азиатский барсук,
- 8 — соболь,
- 9 — амурский лесной кот,
- 10 — манул,
- 11 — рысь,
- 12 — тигр,
- 13 — ирбис,
- 14 — аргали,
- 15 — козерог,
- 16 — волк,

17 — лиса,
18 — енотовидная собака,
19 — заяц,
20 — сурок.

Обратите внимание, что в датасете присутствуют сложные примеры: некоторые виды животных схожи между собой (например, сибирская косуля, пятнистый олень, изюбрь и марал), фото, когда в кадр попала небольшая часть животного, а также ночные и смазанные снимки.

Данные были размечены по видам учеными зоологами. В спорных случаях разметка подтверждалась несколькими экспертами.

Данные разделены на тренировочную и тестовую выборки в соотношении примерно 70% / 30%.

Тестовая часть разбита на публичную и приватную в соотношении примерно 40% / 60%.

Данные: https://disk.yandex.ru/d/Jx8fZJTSsk_LeQ

- `train.zip` — архив с данными для обучения;
- `train.csv` — файл с разметкой для тренировочных данных;
- `test.zip` — тестовые данные, на которых нужно сделать предсказания;
- `baseline.ipynb` — `jupyter-notebook` с простым начальным решением;
- `sample_submission.csv` — пример файла с предсказаниями (все метки классов заполнены нулями).

Baseline (базовое решение)

`Jupyter-notebook`, позволяющий пройти путь от установки библиотек и обучения модели до получения файла с предсказаниями, который можно загрузить на платформу и увидеть метрики.

Формат входных данных

Соревнование подразумевает отправку файла с предсказаниями модели на платформу для расчета метрики. Сам файл представляет собой `csv`-документ с двумя колонками `image_name` и `predicted_class`. Строки — это пары с названием файла изображения и соответствующего ему индексу класса, который предсказала ваша модель.

Пример формирования корректного файла с предсказаниями.

Структура файла с предсказаниями, см. таблицу 3.2.2.

Таблица 3.2.2

`image_name`	`predicted_class`
48daae9b7d2453e77a283be71fc5b2c0.JPG	0
0a74e85fedb477adb3f71c9202bc3ddb.JPG	0
9d5906140ff8487b33e4515a3aff98a4.JPG	0

224b9c7024c0cea46c851f4eb9d9f662.JPG	0
483693f497ace35bbfc60fc4142830f0.JPG	0

Использование сторонних датасетов

Допускается использование открытых (доступных в сети) датасетов с лицензией, позволяющей свободное некоммерческое использование.

Решение

Разрабатывается алгоритм для классификации группы животных по фото с фотоловушек и представить его результат работы в виде `csv`-файла с колонками `image_name`, `predicted_class`.

Решением задачи будет `csv`-файл с предсказанием, загружаемый на платформу проведения соревнования для оценки качества решения. Пример загружаемого файла с решением: `submission.csv`.

Примечания

Разбор эталонного решения доступен по ссылке: https://gist.github.com/ntomaterials/b44ef19161f1dcb4259667d967abfbd2.

4. Заключительный этап

4.1. Работа наставника НТО при подготовке к этапу

На этапе подготовки к заключительному этапу НТО наставник решает две важные задачи: помощь участникам в подготовке к предстоящим соревнованиям и формирование устойчивой и слаженной команды. Заключительный этап требует высокой слаженности, уверенности и глубоких знаний, и наставник становится тем, кто объединяет усилия участников и направляет их в нужное русло.

Наставник помогает участникам:

- разобрать задания прошлых лет, используя официальные сборники, чтобы понять структуру финальных испытаний, типы задач и ожидаемый уровень сложности;
- изучить организационные особенности заключительного этапа, включая формат проведения, регламент, продолжительность и технические нюансы;
- спланировать подготовку — на основе даты начала финала составляется четкий график занятий, в котором распределены темы, практикумы и командные тренировки;
- обратиться (при необходимости) за консультацией к разработчикам заданий по профилю, уточнить, на какие аспекты подготовки следует обратить особое внимание, и получить дополнительные материалы.

Также рекомендуется участие в мероприятиях от организаторов, таких как:

- установочные вебинары и открытые разборы задач;
- хакатоны, практикумы и мастер-классы для финалистов;
- встречи в онлайн-формате, информация о которых публикуется в группе НТО во «ВКонтакте» и в телеграм-чатах профилей.

Наставнику необходимо уделить внимание работе на формированием устойчивой, продуктивной и мотивированной команды:

- **Сплочение команды.** Это особенно актуально, если участники живут в разных городах. Регулярные онлайн-встречи, совместная работа над задачами и неформальное общение помогают наладить доверие и улучшить командную динамику.
- **Анализ ролей.** Наставник вместе с командой определяет, кто за что отвечает, какие задачи входят в зону ответственности каждого участника. Также обсуждаются возможности взаимозаменяемости на случай непредвиденных ситуаций.
- **Оценка компетенций.** Важно определить, какими знаниями и навыками уже обладают участники, а какие необходимо развить. На основе этого формируется индивидуальный и командный план подготовки.
- **Участие в подготовительных мероприятиях от разработчиков профилей.**

Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.

- **Практика в формате хакатонов.** Наставник может организовать дистанционные хакатоны или практикумы с использованием заданий прошлых лет и методических рекомендаций из официальных сборников.

Таким образом, наставник становится координатором и моральной опорой команды, помогая пройти заключительный этап НТО с максимальной уверенностью и результатом.

4.2. Предметный тур

Задачи третьего этапа предметного тура профиля по информатике открыты для решения. Участие в соревновании доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/72666/enter/.

4.2.1. Информатика. 8–11 классы

Задача 4.2.1.1. Игры разума (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одним из развлечений для любого разума является нахождение красивых математических объектов. Один ИИ решил строить пифагоровы треугольники, состоящие из двух других пифагоровых треугольников.

Напомним, что тройка натуральных чисел a, b, c называется пифагоровой, если $a^2 + b^2 = c^2$. Если нарисовать треугольник со сторонами a, b, c, то он получится прямоугольным с гипотенузой длины c и катетами a и b. Самый известный пифагоров треугольник имеет стороны 3, 4, 5.

В пифагоровом треугольнике можно провести высоту из прямого угла на его гипотенузу и получить разбиение на два других прямоугольных треугольника, однако далеко не всегда эти два треугольника будут пифагоровыми. Вот ИИ и задался вопросом: существует ли некоторый пифагоров треугольник, который можно разбить таким образом на два других пифагорова треугольника? И ответ обнадеживает: такие треугольники есть. Пример одного такого треугольника представлен на рис. 4.2.1.

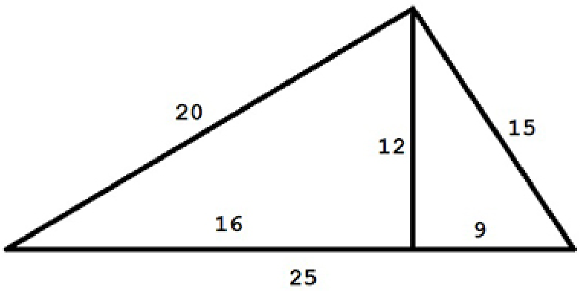


Рис. 4.2.1

Можно видеть, что для всех трех треугольников на этом рисунке выполняются требуемые равенства: $15^2 + 20^2 = 25^2$, $12^2 + 16^2 = 20^2$, $9^2 + 12^2 = 15^2$.

Обозначим стороны самого маленького из трех треугольников в разбиении через a_1, b_1, c_1, стороны среднего — через a_2, b_2, c_2, а стороны самого большого, того, который разбивается на два меньших — через a_3, b_3, c_3. Всегда выполняется $a_i < b_i < c_i$. В примере на рисунке $a_1 = 9$, $b_1 = 12$, $c_1 = 15$, $a_2 = 12$, $b_2 = 16$, $c_2 = 20$, $a_3 = 15$, $b_3 = 20$, $c_3 = 25$.

Теперь ИИ хочет познакомить со своей коллекцией таких треугольников. Необходимо по заданным a_3, b_3, c_3 разбить предлагаемый пифагоров треугольник на два меньших. Гарантируется, что для заданных трех чисел ответ всегда существует.

Формат входных данных

В одну строку через пробел задаются три натуральных числа a_3, b_3, c_3 — стороны большого пифагорова треугольника. Гарантируется, что его можно разбить на два меньших пифагорова треугольника, если провести высоту из прямого угла на его гипотенузу.

$1 \leqslant a_3 < b_3 < c_3 \leqslant 10^9$.

Формат выходных данных

Вывести в две строки стороны двух пифагоровых треугольников, на которые разбивается заданный на входе. В первую строку вывести через пробел стороны a_1, b_1, c_1 самого маленького треугольника разбиения, а во вторую сторону — стороны среднего. Всегда должно выполняться $a_i < b_i < c_i$ и $a_1 < a_2, b_1 < b_2, c_1 < c_2$.

Примеры

Пример №1

Стандартный ввод
15 20 25

Стандартный вывод
9 12 15 12 16 20

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
```

```
    int a3, b3, c3;
    cin >> a3 >> b3 >> c3;
    int a1 = a3 * a3 / c3;
    int b1 = a3 * b3 / c3;
    int c1 = a3;
    int a2 = b1;
    int b2 = b3 * b3 / c3;
    int c2 = b3;
    cout << a1 <<' '<< b1<<' '<< c1<< endl;
    cout << a2 <<' '<< b2<<' '<< c2<< endl;
}
```

Задача 4.2.1.2. Восстановление сообщения (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим следующую задачу из области криптографии.

Алиса посылает Бобу сообщение, которое состоит из n **положительных** целых чисел a_i (i от 1 до n). Злоумышленница Мэллори перехватывает эти числа и каждое a_i преобразует в какое-то также **положительное целое** число t_i по следующему правилу: изначально у Мэллори есть число $t_0 = 0$. Далее, перехватив очередное число Алисы a_i, Мэллори из него получает число t_i. Если $a_i \leqslant t_{i-1}$, то $t_i = a_i + t_{i-1}$, если же $a_i > t_{i-1}$, то Мэллори может случайным образом выбрать t_i из двух вариантов $t_i = a_i + t_{i-1}$ или $t_i = a_i - t_{i-1}$. После этого полученное t_i отправляется Бобу, а также используется для получения следующего t_{i+1}.

Допустим, если Алиса хочет переслать числа $2, 6, 5, 8, 11, 1$, то Мэллори может сделать, например, следующее преобразование (в суммах первое число — это a_i, второе число t_{i-1}) — $t_1 = 2 + 0 = 2$, $t_2 = 6 - 2 = 4$, $t_3 = 5 - 4 = 1$, $t_4 = 8 + 1 = 9$, $t_5 = 11 - 9 = 2$, $t_6 = 1 + 2 = 3$. После этого Боб получит сообщение $2, 4, 1, 9, 2, 3$.

Алиса, зная, что при передаче сообщения возможны взломы, решила продублировать его два раза подряд, то есть Мэллори, не зная этого, будет преобразовывать удвоенное сообщение. Боб, получив сообщение и зная проделки Мэллори, хочет восстановить исходное сообщение Алисы, для чего использует ИИ.

В решении задачи нужно не отстать от ИИ и написать программу, которая восстановит исходное сообщение Алисы вне зависимости от зловредных действий Мэллори. Можно показать, что при указанным условиях любое удвоенное сообщение можно восстановить однозначно.

Для лучшего понимания условия рекомендуем разобрать тесты из условия, которые получились из одного и того же сообщения Алисы, и иллюстрируют разные варианты выбора Мэллори в тех случаях, когда она может как складывать, так и вычитать.

Формат входных данных

В первой строке подается одно целое число n — количество чисел в исходном сообщении Алисы (до того, как она его удвоила), $1 \leqslant n \leqslant 10^5$.

Во второй строке через пробел приведены $2 \cdot n$ чисел t_i — те, что сгенерировала Мэллори и получил Боб, $1 \leqslant t_i \leqslant 10^9$. Гарантируется, что эти $2 \cdot n$ чисел t_i получены из некоторого удвоенного исходного набора Алисы по правилам из условия.

Формат выходных данных

Вывести n чисел — восстановленное сообщение, исходно созданное Алисой до того, как она его удвоила.

Примеры

Пример №1

Стандартный ввод
6 2 4 1 9 2 3 5 1 6 2 9 10

Стандартный вывод
2 6 5 8 11 1

Пример №2

Стандартный ввод
6 2 4 1 7 4 5 7 13 18 26 37 38

Стандартный вывод
2 6 5 8 11 1

Пример №3

Стандартный ввод
6 2 8 13 21 32 33 35 41 46 54 65 66

Стандартный вывод
2 6 5 8 11 1

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;
signed main(){
    int n;
    cin >> n;
    vi v(2 * n);
    for0(i, 2 * n){
        cin >> v[i];
    }
    int t = 0;
    vector<pii> tmp(2 * n, {0, 0});
    for0(i, 2 * n){
        tmp[i].x = v[i] + t;
        if(v[i] > t){
            tmp[i].y = v[i] - t;
        }
        t = v[i];
    }
    for0(i, n){
        int tans;
        if(tmp[i].x == tmp[i + n].x || tmp[i].x == tmp[i + n].y){
            tans = tmp[i].x;
        }
        else{
            tans = tmp[i].y;
        }
        cout << tans <<' ';
    }
    cou

```

Задача 4.2.1.3. Высотная поясность (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Группа биологов и ботаников исследует малоизвестный горный район. Каждый день они перемещаются по склону горного массива, фотографируют, описывают, проводят исследования и т. п. За один день группа либо поднимается на один дневной переход вверх, либо опускается также на один дневной переход вниз. Упрощенно можно считать, что расстояния, проходимые группой в течение любого дня, одинаковы, а угол к горизонту, под которым группа поднимается или опускается, равен $45°$.

Исследователями был подмечен следующий факт: каждый раз место остановки группы на ночлег находилось как раз на границе зоны высотной поясности. Например, в первый день они поднимались по тропическому лесу, во второй — прошли пояс хвойных лесов, в третий — поднялись в пояс альпийских лугов, в четвертый — снова спустились в пояс хвойных лесов и т. д. В итоге группа последним переходом спустилась точно на ту же высоту h_0, с которой начала маршрут. Помимо этого, известно, что во время прохождения маршрута группа не опускалась ниже высоты h_0, но в промежуточных точках могла останавливаться на этой высоте.

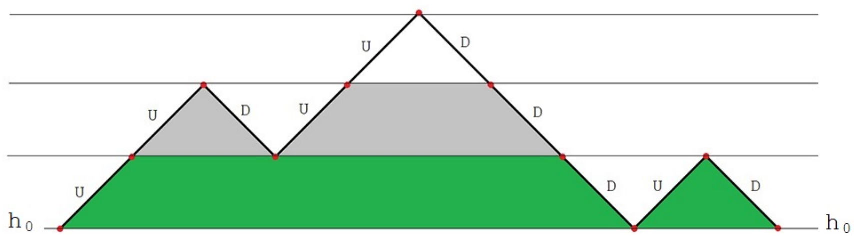


Рис. 4.2.2

Пришло время обрабатывать результаты. В числе прочих руководитель экспедиции хочет начертить двумерную диаграмму, схематично представляющую пройденный маршрут, и по ней оценить площадь каждого пояса горного рельефа, пройденного группой.

По описанию пройденного группой маршрута вывести площадь каждого пояса на полученной двумерной диаграмме.

Формат входных данных

В первой строке приведено целое четное число n — количество дневных переходов, сделанных группой, $2 \leqslant n \leqslant 2 \cdot 10^5$.

Во второй строке содержится набор из n символов U и D, описывающий характер движения группы в каждый из дней. Символ U означает, что в соответствующий день группа поднималась, а символ D означает, что в этот день группа опускалась. Гарантируется, что для этого маршрута выполняются все условия, приведенные в описании задачи.

Формат выходных данных

В первой строке следует вывести число k — количество высотных поясов, которые посетила группа.

В следующей строке вывести k чисел через пробел — площади каждого пояса на двумерной диаграмме маршрута. Пояса нумеруются с самого нижнего до самого верхнего по возрастанию высот.

В качестве единицы измерения следует использовать проекцию одного дневного перехода на горизонтальную ось и считать, что длина одного дневного перехода равна $\sqrt{2}$. Можно показать, что площадь любого пояса в такой системе будет целым числом.

Примеры

Пример №1

Стандартный ввод
10 UUDUUDDDUD

Стандартный вывод
3 8 4 1

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int> > vvi;
typedef vector<double> vd;
typedef vector<vd> vvd;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;
signed main(){
    int n;
    cin >> n;
    string s;
    cin >> s;
    vi st;
    vi ans;
    for0(i, n){
        if(s[i] == 'U'){
            st.pb(i);
            if(sz(ans) < sz(st)){
                ans.pb(0);
            }
            continue;
```

```
        }
        int d = i - st.back();
        ans[sz(st) - 1] += d;

        st.pop_back();
    }
    cout << sz(ans) << endl;
    for0(i, sz(ans)){
        cout << ans[i] <<' ';
    }
    cout << endl;
}
```

Задача 4.2.1.4. Электромобиль (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 128 Мбайт.

Условие

Иван Иванович купил электромобиль и хочет поехать на нем в путешествие по прямой. Емкость аккумулятора его электромобиля 100 кВтч. На один километр его машина тратит 1 кВтч. Так как электромобиль произведен неизвестным широкой публике производителем, есть небольшое условие: его аккумулятор может за одну зарядку увеличить запас только на количество киловатт-часов, равное квадрату натурального числа, то есть на 1, 4, 9, 16, 25, 36, 49, 64, 81 или 100 кВтч, при условии, что эта добавка не превысит общую емкость аккумулятора в 100 кВтч.

Например, если в момент начала зарядки аккумулятор хранил 60 кВтч, то после зарядки он может содержать 61, 64, 69, 76, 85, или 96 кВтч.

Иван Иванович знает, что по пути он встретит n электрозаправочных станций. До каждой из этих станций известно расстояние в километрах от начала маршрута, равное d_i. Иван Иванович может как угодно действовать в рамках описанных выше ограничений: пропустить заправку вообще, либо заправить на ней любое разрешенное условием квадратное число киловатт-часов. На каждой станции он может заряжать автомобиль только один раз. Исходно аккумулятор его электромобиля был полностью заряжен.

Иван Иванович поставил ИИ, управляющему электромобилем, задание составить режим дозаправок по этим данным так, чтобы проехать максимальное расстояние от начала маршрута. Если в момент приезда на заправку аккумулятор опустошается и содержит 0 кВтч, то Иван Иванович может заправить любое количество, равное квадрату натурального числа от 1 до 100 кВтч и продолжать двигаться далее. В конце маршрута Иван Иванович планирует вызвать эвакуатор, поэтому будет продолжать движение до тех пор, пока не опустеет аккумулятор.

Формат входных данных

В первой строке содержится одно число n — количество электрозаправок на пути Ивана Ивановича, $0 \leqslant n \leqslant 10^5$.

В следующей строке содержатся n целых чисел d_i через пробел — расстояние от точки старта до i-й заправки, $1 \leqslant d_i \leqslant 10^7$. Заправки описаны в порядке увеличения расстояния от точки старта, то есть $d_1 < d_2 < \ldots < d_n$. Никакие две заправки не находятся в одной точке.

Формат выходных данных

Вывести одно число — максимальное расстояние, на которое сможет уехать Иван Иванович от точки старта.

Примеры

Пример №1

Стандартный ввод
`4` `45 101 196 224`

Стандартный вывод
`318`

Примечания

В примере из условия одна из оптимальных стратегий следующая:

1. Проехать от точки старта до первой заправки 45 км, останется заряд 55 кВтч. На первой заправке заправить 1 кВтч, теперь заряд 56 кВтч.
2. Проехать от первой заправки до второй заправки 56 км, останется заряд 0 кВтч. На второй заправке заправить 100 кВтч, теперь заряд 100 кВтч.
3. Проехать от второй заправки до третьей заправки 95 км, останется заряд 5 кВтч. На третьей заправке заправить 36 кВтч, теперь заряд 41 кВтч.
4. Проехать от третьей заправки до четвертой заправки 28 км, останется заряд 13 кВтч. На четвертой заправке заправить 81 кВтч, теперь заряд 94 кВтч.
5. Проехать от четвертой заправки до конца маршрута 94 километра, заряд 0 кВтч.

Окончание маршрута в точке 318 км. Можно показать, что это наилучший возможный для первого примера результат.

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#pragma GCC target("popcnt")
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int> > vvi;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;
signed main(){
        ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    int n;
    cin >> n;
    if(n == 0){
        cout << 100 << endl;
        return 0;
    }
    vector<bitset<101> > sq(101);
    for0(i, 101){
        for1(j, 10){
            if(i + j * j <= 100){
                sq[i][i + j * j] = 1;
            }
        }
    }
    bitset<101> u = sq[99];
    int ans = 0;
    if(u.count() != 0){
        for(int j = 100; j >= 0; j--){
            if(u[j]){
                ans = max(ans, j);
                break;
            }
        }
    }
    int p = 0;
    for0(i, n){
        int d;
        cin >> d;
        u = (u >> (d - p));
        p = d;
        bitset<101> dp = 0;
        for0(j, 101){
            if(u[j]){
                dp |= sq[j];
            }
        }
        u |= dp;
        if(u.count() != 0){
            for(int j = 100; j >= 0; j--){
                if(u[j]){
```

```
                    ans = max(ans, d + j);
                    break;
                }
            }
        }
    }
    cout << ans << endl;
}
```

Задача 4.2.1.5. Перелетные кузнечики (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

На далекой планете был открыт гигантский архипелаг, заселенный исключительно кузнечиками. Исследователи заметили, что эти кузнечики были p различных видов. На каждом острове архипелага проживал ровно один вид кузнечиков, при этом некоторые их виды захватили множество островов. Помимо естественной возможности передвигаться обычным образом по своему острову, кузнечики обладали сверхспособностью совершать гигантские суперпрыжки с острова на остров. Для этого они пользовались направлениями магнитного поля, которое на этой планете позволяло делать прыжки строго в меридиональном направлении (север-юг или юг-север), или строго в направлении параллели (запад-восток или восток-запад) и никак иначе.

Кузнечики могут передвигаться обычным образом по суше внутри любого единичного квадрата и между двумя рядом расположенными единичными квадратами, имеющими общую сторону. Если два единичных квадрата расположены рядом, но по диагонали и между ними нет пути через клетки с общими сторонами, это два различных острова, напрямую между ними передвижение невозможно.

Для исследования вопроса происхождения и расселения этих видов и объяснения возникшей ситуации был подключен ИИ, который сделал предположение, что изначально каждый вид кузнечиков зародился на своем острове и потом они, пользуясь своей способностью к суперпрыжкам, расселились по всему архипелагу. Если у двух островов есть единичные квадраты, расположенные в одном столбце или строке, то между ними возможно передвижение при помощи суперпрыжка, поэтому на данных двух островах обязательно проживают кузнечики одного вида.

Теперь у ученых возник новый вопрос: для каждого вида они хотят знать минимальную необходимую длину суперпрыжка, обладая которой, этот вид смог расселиться по островам, которые сейчас заселяет.

По заданной карте архипелага требуется определить число видов кузнечиков, обитающих на архипелаге, и вывести для каждого вида минимальную длину суперпрыжка, при помощи которого этот вид смог заселить свои острова. Расстояние суперпрыжка между двумя единичными клетками вычисляется как расстояние между центрами этих клеток.

Формат входных данных

В первой строке содержатся числа n и m — размеры архипелага, и далее в следующих n строках представлена карта архипелага в виде таблицы $n \times m$, каждая ячейка которой либо принадлежит острову, и тогда в ней содержится 1, либо морю, тогда в ней содержится 0.

Числа n и m имеют следующие ограничения: $5 \leqslant n, m \leqslant 500$.

Формат выходных данных

В первой строке следует вывести одно число p — количество видов кузнечиков.

В следующих строке требуется вывести p чисел через пробел. Для каждого вида нужно вывести его минимальную возможную длину суперпрыжка. Виды нумеруются и перечисляются по самой северной строке их обитания. Более строго, среди всех точек обитания вида находится самая северная (самая верхняя на карте). Так как в этой строке будут расположены только точки обитания этого вида кузнечиков, то каждый вид однозначно определяется по самой северной строке своего обитания. Из двух видов первым будет выведен ответ для того вида, у которого эта строка будет встречена первой при движении сверху вниз.

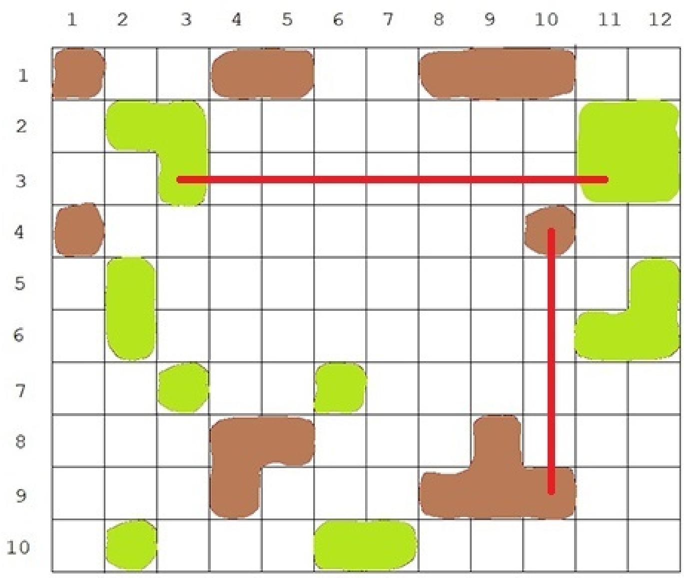


Рис. 4.2.3

На рис. 4.2.3 приведена карта архипелага из примера № 1. Острова на ней уже раскрашены согласно обитающим на них видам кузнечиков. Можно видеть, что в данном случае возникло два вида кузнечиков. Первый из них, ареал обитания которого раскрашен коричневым, имеет самую северную точку обитания в строке 1 и должен обладать длиной суперпрыжка как минимум 5, а второй, ареал которого раскрашен зеленым, имеет самую северную точку обитания в строке 2 и должен обладать длиной суперпрыжка как минимум 8. Соответствующие прыжки указаны на рисунке.

Примеры

Пример №1

Стандартный ввод

```
10 12
100110011100
011000000011
001000000011
100000000100
010000000001
010000000011
001001000000
000110001000
000100011100
010001100000
```

Стандартный вывод

```
2
5 8
```

Пример №2

Стандартный ввод

```
10 12
100111011100
011000000011
001000000011
100000000100
010000000001
010000000011
001001000000
000110001000
000100011100
010001100001
```

Стандартный вывод

```
1
6
```

Пример №3

Стандартный ввод
8 8 11100011 10000110 10000100 00010000 00001000 00000001 00000011 00100000

Стандартный вывод
3 7 0 0

Пример №4

Стандартный ввод
8 8 11100011 10000110 10000100 00011000 00011000 00000001 00000011 00100100

Стандартный вывод
2 5 0

Пример №5

Стандартный ввод
5 5 11111 10000 11111 00001 11111

Стандартный вывод
1 0

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int> > vvi;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;

int n, m;
int dx[4] = {-1, 0, 1, 0};
int dy[4] = {0, 1, 0, -1};
bool inside(int x, int y){
    return (x >= 0 && y >= 0 && x < n && y < m);
}

vector<string> T;
vvi col;
vector<vector<pii> > G;
vector<pii> st1;
void nodfs1(int x, int y, int c){
    col[x][y] = c;
    st1.pb({x, y});

    while(sz(st1) > 0){
        pii top = st1.back();
        st1.pop_back();
        for0(i, 4){
            int nx = top.x + dx[i];
            int ny = top.y + dy[i];
            if(inside(nx, ny) && T[nx][ny] == '1' && !col[nx][ny]){
                st1.pb({nx, ny});
                col[nx][ny] = c;
            }
        }
    }
}

vi used;
vvi lst;
vi emp;
vi st2;
void nodfs2(int a){
    st2.pb(a);
    used[a] = 1;
    lst.back().pb(a);
    while(sz(st2) > 0){
        int top = st2.back();
        st2.pop_back();
        for(auto to : G[top]){
            if(!used[to.x]){
```

```
                st2.pb(to.x);
                used[to.x] = 1;
                lst.back().pb(to.x);
            }
        }
    }
}

vi dist;
int prim(int num){
    int res = 0;
    dist[lst[num][0]] = 0;
    set<pii> que;
    que.insert({0, lst[num][0]});
    for(int i = 1; i < sz(lst[num]); i++){
        que.insert({INF, lst[num][i]});
    }
    for(int u = 0; u < sz(lst[num]); u++){
        pii top = *que.begin();
        used[top.y] = 1;
        que.erase(que.begin());
        res = max(res, top.x);
        for(auto to : G[top.y]){
            if(used[to.x]){
                continue;
            }
            que.erase({dist[to.x], to.x});
            dist[to.x] = min(dist[to.x], to.y);
            que.insert({dist[to.x], to.x});
        }
    }
    return res;
}

signed main(){
        ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n >> m;
    T.resize(n);
    col.resize(n, vi(m, 0));

    for0(i, n){
        cin >> T[i];
    }
    int z = 0;
    for0(i, n){
        for0(j, m){
            if(T[i][j] == '1' && !col[i][j]){
                z++;
                nodfs1(i, j, z);
            }
        }
    }
    G.resize(z + 1);
    for0(i, n){
        int last = -1, d = 0;
        for0(j, m){
            if(last == -1){
                if(col[i][j] != 0){
                    last = col[i][j];
                }
```

```
                continue;
            }
            if(col[i][j] == 0){
                d++;
                continue;
            }
            if(col[i][j] != 0){
                if(last != col[i][j]){
                    G[last].pb({col[i][j], d + 1});
                    G[col[i][j]].pb({last, d + 1});
                }
                    last = col[i][j];
                    d = 0;
            }
        }
    }
    for0(j, m){
        int last = -1, d = 0;
        for0(i, n){
            if(last == -1){
                if(col[i][j] != 0){
                    last = col[i][j];
                }
                continue;
            }
            if(col[i][j] == 0){
                d++;
                continue;
            }
            if(col[i][j] != 0){
                if(last != col[i][j]){
                    G[last].pb({col[i][j], d + 1});
                    G[col[i][j]].pb({last, d + 1});
                }
                    last = col[i][j];
                    d = 0;
            }
        }
    }
    for1(i, z){
        sort(all(G[i]));
    }
    vector<pii> tmp;
    for1(i, z){
        tmp.resize(0);
        tmp = G[i];
        G[i].resize(0);
        if(sz(tmp) > 0){
            G[i].pb(tmp[0]);
            for1(j, sz(tmp) - 1){
                if(tmp[j].x != tmp[j - 1].x){
                    G[i].pb(tmp[j]);
                }
            }
        }
    }
    used.resize(z + 1, 0);
    for1(i, z){
        if(!used[i]){
            lst.pb(emp);
```

```
                nodfs2(i);
            }
        }
        dist.resize(z + 1, INF);
        used.resize(0);
        used.resize(z + 1, 0);
        cout << sz(lst) << endl;
        for0(i, sz(lst)){
            cout << prim(i) << ' ';
        }
        cout << endl;
    }
```

4.2.2. Математика. 8–9 классы

Задача 4.2.2.1. (15 баллов)

Тема: *алгебра и теория чисел.*

Условие

Даны три натуральных числа a, b, c, удовлетворяющие следующим условиям:

- $a > 1$,
- ab — простое число,
- bc — произведение двух простых чисел,
- abc не кратно квадратам простых чисел,
- $abc \leqslant 42$.

Найдите все тройки чисел (a, b, c), удовлетворяющих указанным условиям.

Решение

Так как $a > 1$, ab — простое, то $b = 1$ и a — простое.

Из третьего условия $bc = 1 \cdot c = c$. Значит, c — это произведение двух различных простых чисел $p_1 \cdot p_2$, так как если $p_1 = p_2 = p$, то abc делится на p^2, что недопустимо по условию. Тогда можно считать, что abc — это произведение трех различных простых чисел $abc = p_1 \cdot p_2 \cdot p_3$.

Заметим, что наименьшие произведения дают тройки $2 \cdot 3 \cdot 5 = 30$, $2 \cdot 3 \ \cdot 7 = 42$. Все остальные произведения трех простых чисел будут больше 42. Тогда возможны следующие наборы чисел (a, b, c): (2, 1, 15), (3, 1, 10), (5, 1, 6), (2, 1, 21), (3, 1, 14), (7, 1, 6).

Ответ: (2, 1, 15), (3, 1, 10), (5, 1, 6), (2, 1, 21), (3, 1, 14), (7, 1, 6).

Критерии оценивания

Приведен только верный ответ без объяснений — 5 баллов.

Потеряна одна из серий решений, то есть получено только 3 ответа — 7 баллов.

Нет объяснений, почему простые числа должны быть различными, хотя это подразумевается в решении — снять 2 балла.

Потерян один из ответов при верном решении — 12 баллов.

Отсутствуют объяснения, что нет других подходящих троек простых чисел, кроме 2, 3, 5 и 2, 3, 7 — баллы не снижать.

Задача 4.2.2.2. (20 баллов)

Темы: *комбинаторика, множества.*

Условие

Пусть множество S является наименьшим набором натуральных чисел, удовлетворяющих следующим свойствам:

- число 2 принадлежит S,
- если n^2 принадлежит S, то n также принадлежит S,
- если n принадлежит S, то $(n+5)^2$ также принадлежит S.

1. Верно ли, что число $4\underbrace{0\ldots0}_{2025}4$ принадлежит S? (в этом числе 2025 нулей).
2. Найдите все числа $n \geqslant 10\,000$, которые принадлежат S.

Решение

Если $n \in S$, то $(n+5)^2 \in S \Longrightarrow n+5 \in S$, тогда для любого натурального k верно: $n + 5k \in S$.

Так как $2 \in S \Longrightarrow (2+5)^2 = 49 \in S$, кроме того, $5k+2 \in S$ для любого целого $k \geqslant 0$.

Из того, что $49 \in S$ следует, что все числа вида $5k+4 \in S$, где $k \geqslant 9$.

Отсюда сразу получаем решение для п.1, так как число $4\underbrace{0\ldots0}_{2025}4$ имеет вид $5k+4$.

Из того, что $49 \in S \Longrightarrow (49+5)^2 = 54^2 = 2916 \in S$, значит, все числа вида $5k+1 \in S$, где $k \geqslant 583$.

Так как $54 \in S \Longrightarrow 54+2\cdot5 = 64 \in S \Longrightarrow 8 \in S$, значит, все числа вида $5k+3 \in S$, где $k \geqslant 1$.

В итоге получили, что все числа $n \geqslant 10\,000$, дающие остатки 1, 2, 3, 4 при делении на 5, содержатся в S.

Покажем теперь, что никакое число, кратное 5, не может содержаться в S. Действительно, n^2 делится на 5 тогда и только тогда, когда n делится на 5. А также $(n+5)^2$ делится на 5 тогда и только тогда, когда n делится на 5. Но изначально в множестве S есть только число 2, которое не делится на 5, поэтому нельзя получить никакое число, кратное 5.

Замечание. Можно доказать более общее утверждение: в S содержатся все натуральные числа, кроме 1 и чисел, кратных 5.

Так как $54 = 49+5 \in S \Longrightarrow 54^2 \in S$. Заметим, что числа вида $(5k+4)^2 \equiv 1 \pmod 5$, значит, все числа вида $5k+1$, большие 54^2, содержатся в S. Значит, $256^2 \in S$. Отсюда получаем, что $256 \in S$, $16 = \sqrt{256} \in S, 4 = \sqrt{16} \in S$. Следовательно, любое число вида $5k+4 \in S$.

Далее $4 \in S \Longrightarrow 4+5 = 9 \in S \Longrightarrow 3 \in S$. Поэтому любое число вида $5k+3 \in S$.

Так как $16 \in S$, то $16 + 5\cdot 4 = 36 \in S \Longrightarrow 6 \in S$. Значит, все числа вида $5k+1$, начиная с 6, содержатся в S.

Осталось показать, что 1 не может содержаться в S. Это следует из того, что $1 = 1^2$, а значит, 1 нельзя получить из чисел, больших 1.

Ответ: 1. Да. 2. Все натуральные числа, кроме кратных 5.

Критерии оценивания

Приведен только ответ в пункте 1 — 0 баллов.

Приведен только ответ в пункте 2 — 1 балл.

Доказано, что все числа вида $5k+i \geqslant 10\,000$ содержатся в S — добавить 2 балла для каждого i.

Доказано, что числа, кратные 5, не содержатся в S — 5 баллов.

Верно решен пункт 1 — 7 баллов.

При решении пункта 2 доказано, что любое число $n \geqslant 10\,000$, некратное 5, подходит — 8 баллов.

Задача 4.2.2.3. (20 баллов)

Темы: комбинаторика, игры и стратегии.

Условие

Заданы целые числа $m > n \geqslant 3$. Рамкой прямоугольника $m \times n$ называется фигура, состоящая из $2m+2n-4$ граничных квадратов этого прямоугольника. На рис. 4.2.4 показана рамка размером 4×7 в качестве примера.

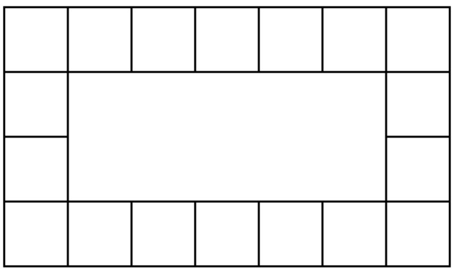


Рис. 4.2.4

На рамке размера $m \times n$, все клетки которой первоначально белые, играют робот А и робот Б. Ходят по очереди, первый ход делает А. За один ход нужно выбрать полностью белый прямоугольник $1 \times k$, $k \geqslant 1$ и перекрасить все его клетки в черный цвет так, чтобы оставшиеся белые клетки образовывали связную фигуру (клетки, имеющие лишь общую вершину, связными не являются). Выиграет игрок, сделавший последний ход.

Найдите все пары (m, n), для которых у робота А есть выигрышная стратегия, позволяющая ему победить вне зависимости от ходов противника.

Решение

Пусть для определенности левая и правая границы рамки состоят из n клеток, а верхняя и нижняя — из m клеток.

Приведем выигрышную стратегию робота А и покажем, что он всегда может ее придерживаться.

Робот А первым своим ходом закрашивает нижнюю границу рамки (прямоугольник $1 \times m$).

Теперь у робота Б есть возможность закрасить клетки только на левой или правой границе. Следующими ходами робот А закрашивает столько же клеток, сколько закрасил Б, но на противоположной границе. Например, Б закрасил k клеток на левой границе, тогда А закрашивает k клеток на правой. Так продолжается до тех пор, пока одна из границ не будет полностью закрашена. Заметим, что это обязательно произойдет после хода робота Б. Пусть теперь осталось s незакрашенных клеток на боковой границе и t на верхней (одна угловая клетка общая). Причем $t > s$, так как боковая граница содержит меньше клеток, чем верхняя. Теперь робот А закрашивает клетки на верхней границе так, чтобы количества оставшихся клеток на верхней и боковой границах было одинаковым. Далее робот А повторяет ходы робота Б симметрично незакрашенной угловой клетке до тех пор, пока незакрашенная фигура не является прямоугольником. В силу симметрии незакрашенная фигура станет прямоугольником после хода робота Б. После этого робот А закрасит оставшийся прямоугольник и выиграет.

Ответ: робот А выигрывает для любых m, n.

Критерии оценивания

Указано, что робот А всегда выигрывает без объяснений — 0 баллов.

Приведена верная стратегия робота А, однако не доказано, что ее всегда можно реализовать — 15 баллов.

Задача 4.2.2.4. (20 баллов)

Тема: геометрия.

Условие

Две окружности ω_1 и ω_2 пересекаются в двух различных точках A и B. Прямая, проходящая через B, снова пересекает ω_1 и ω_2 в точках C и D соответственно. Докажите, что если $CA = CD$, то центр ω_2 лежит на окружности ω_1.

Решение

Проведем перпендикуляр из точки C на AD. Он пересечет ω_1 в некоторой точке O. Поскольку $CA = CD$, то CO является серединным перпендикуляром к AD, а также биссектрисой угла $\angle ACD$. Тогда треугольники AOC и DOC равны, значит,

$OA = OD$. Так $\angle OCB = \angle OAB$ как вписанные углы, опирающиеся на дугу $\overset{\frown}{OB}$, а $\angle ACO = \angle ABO$ как вписанные углы, опирающиеся на дугу $\overset{\frown}{AO}$, то треугольник AOB — равнобедренный.

Таким образом, получаем $OA = OB = OD$. Это означает, что O равноудалена от вершин A, B, D и является центром описанной окружности треугольника ADB, а значит, и центром окружности ω_2.

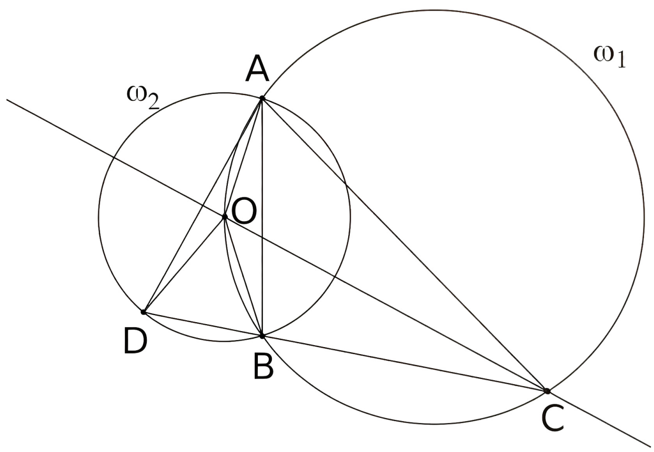


Рис. 4.2.5

Критерии оценивания

Сформулирована идея выбора точки O на биссектрисе угла C — 5 баллов.

Доказано, что $OA = OD$ при выборе точки O на биссектрисе — 10 баллов.

Доказано, что $OA = OB$ при выборе точки O на биссектрисе — 10 баллов.

Задача 4.2.2.5. (25 баллов)

Тема: *алгебра.*

Условие

Пусть $P(x) = x^2 + ax + b$ — квадратный трехчлен. Известно, что числа $P(-1)^2$, $P(0)^2$, $P(1)^2$, являются целыми и в указанном порядке образуют арифметическую прогрессию. Докажите, что a и b — целые числа.

Решение

Так как $P(-1) = 1 - a + b$, $P(0) = b$, $P(1) = 1 + a + b$, то по свойству арифметической прогрессии $2b^2 = (1 - a + b)^2 + (1 + a + b)^2 = 2 + 4b + 2b^2 + 2a^2$. Следовательно, $a^2 + 2b + 1 = 0$.

Заметим, что $1 + a^2 + b^2 + 2a + 2b + 2ab = (1 + a + b)^2 \in \mathbb{Z}$. Так как $1, b^2, a^2 + 2b$ — целые числа, то $2a + 2ab = 2a(1 + b)$ тоже целое, значит, $4a^2(1 + b)^2$ — целое. Но $a^2 = -(2b + 1)$, поэтому $4(2b + 1)(1 + b)^2$ тоже целое.

Так как $4(2b + 1)(1 + b)^2 = 4(1 + 4b + 5b^2 + 2b^3) = 4b(4 + 2b^2) + 4 + 20b^2$, то $4b(4 + 2b^2)$ — целое число. Если $b = 0$, то b — целое, иначе оно рационально, но так как b^2 целое, а квадрат рационального нецелого не может быть целым, значит, b может быть только целым.

Так как $a^2 = -(2b + 1)$, то a^2 — целое. Далее, так как $2a(1 + b)$ — целое, то если $b \neq -1$, получим, что a рациональное, а значит, целое. Если $b = -1$, то $P(-1) = -a$, $P(0) = b = -1$ и $P(1) = a$. Поэтому числа $a^2, 1, a^2$ должны образовывать арифметическую прогрессию. Это означает, что $a^2 = 1$, $a = \pm 1$.

Во всех случаях получаем, что a, b являются целыми числами.

Критерии оценивания

Доказано, что b — целое — 15 баллов.

Доказано, что b — рациональное — 10 баллов.

Доказано, что b — целое и a^2 — целое — 20 баллов.

Упущены случаи такие как, $b = 0, b = -1$ — снять 5 баллов.

4.2.3. Математика. 10–11 классы

Задача 4.2.3.1. (15 баллов)

Темы: комбинаторика, теория чисел.

Условие

Нейросеть получила на выходе упорядоченный по возрастанию список четырехзначных чисел, каждое из которых удовлетворяет следующим условиям:

- в списке только четные числа;
- у каждого числа ровно 3 нечетные цифры и все они различны;
- сумма цифр каждого числа равна 23.

Ответьте на вопросы:

1. Какое первое число записано в списке?
2. Сколько всего чисел в списке?

Решение

Обратите внимание: единственная четная цифра стоит на четвертом месте, так как число четное.

1. Заметим, что $1\,598$ удовлетворяет указанным условиям. Предположим, что есть меньшее число. Его первая цифра не может быть меньше 1. Предположим, что его вторая цифра меньше 5, тогда его первые две цифры — 13. Следовательно, сумма всех его цифр не больше $1+3+8+9=21$ (8 и 9 — максимально возможные цифры). Противоречие.

   Следовательно, первые две цифры данного числа — 15 (иначе число больше $1\,598$). Заметим, что $23-1-5=17$, следовательно, последние две цифры нашего числа — 8 и 9, причем 8 стоит на четвертом месте.

2. Рассмотрим какое-нибудь четырехзначное число, удовлетворяющее указанным условиям. Обозначим через d его последнюю цифру.

   Максимальная сумма трех различных нечетных цифр равна $9+7+5=21$, поэтому $d \geqslant 2$.

   Рассмотрим теперь возможные значения d:

   - Если $d=2$, то сумма трех других цифр равна 21. Единственная возможность 5, 7, 9.
   - Если $d=4$, то сумма трех других цифр равна 19. Единственная возможность 3, 7, 9.
   - Если $d=6$, то сумма трех других цифр равна 17. Возможности 1, 7, 9 или 3, 5, 9.
   - Если $d=8$, то сумма трех других цифр равна 15. Возможности 3, 5, 7 или 1, 5, 9.

   Получаем 6 вариантов. В каждом варианте первые три цифры можно переставлять местами 3! способами. Окончательно получаем $6 \cdot 3! = 36$ чисел.

Ответ: 1. $1\,598$, 2. 36.

Критерии оценивания

Ответ в пункте 1, возможно, без обоснования — 5 баллов.

Верное решение пункта 2 — 10 баллов.

Упущен один из 6 случаев троек нечетных цифр — 5 баллов за пункт 2.

Неверно посчитали число перестановок из трех нечетных цифр (например, 3, вместо 3!) — 5 баллов за пункт 2.

Задача 4.2.3.2. (20 баллов)

Тема: комбинаторика.

Условие

В таблице $N \times N$ каждая из клеток раскрашена в черный или белый цвет. Оказалось, что для любой пары столбцов и любой пары строк среди четырех клеток на их пересечении встречаются хотя бы одна черная и хотя бы одна белая. Какое наибольшее значение может принимать число N?

Решение

Пусть $N \geqslant 5$. Выделим квадрат 5×5. Если его столбец содержит хотя бы три черные клетки, то назовем его черным, в противном случае — белым. Из пяти столбцов хотя бы три — одного цвета. Рассмотрим эти три столбца. Без ограничения общности будем считать их черными. Обозначим строки a, b, c, d, e в каком-либо порядке. Пусть в первом столбце черные клетки стоят в строках a, b, c. Тогда во втором и третьем столбцах максимум по одной черной клетке в этих строках. Значит, во втором и третьем столбцах черные клетки точно стоят в строках d и e, и найдена пара строк d, e и пара столбцов $2, 3$, в пересечении которых все клетки черные. Противоречие.

Пример для $N = 4$ приведен на рис. 4.2.6.

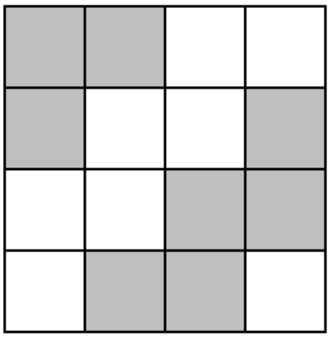


Рис. 4.2.6

Ответ: $N = 4$.

Критерии оценивания

Только верный ответ — 0 баллов.

Дан верный ответ и приведен пример — 5 баллов.

Доказано, что $N \geqslant 5$ не подходит — 15 баллов.

Баллы, указанные в критериях выше, складываются.

Вместо $N \geqslant 5$ рассматривается лишь случай $N = 5$ — баллы не снимать.

Задача 4.2.3.3. (20 баллов)

Тема: алгебра и теория чисел.

Условие

Найдите все тройки целых чисел a, b, c для которых выполняются оба равенства $a^2 = bc + 4$, $b^2 = ac + 4$.

Решение

Пусть $a = b$, тогда $a^2 = ac + 4$ или $a(a - c) = 4$. Получаем, что a равно одному из чисел $\pm 4, \pm 2, \pm 1$. Рассматривая каждый случай, получаем решения $(4, 4, 3)$, $(-4, -4, -3)$, $(2, 2, 0)$, $(-2, -2, 0)$, $(1, 1, -3)$, $(-1, -1, 3)$.

Пусть $a \neq b$. Вычтем из первого уравнения второе $a^2 - b^2 = c(b - a)$. Разделим обе части уравнения на $b - a$ и получим $-a - b = c$. Подставим в первое уравнение $a^2 = -ab - b^2 + 4$. Откуда $2a^2 + 2b^2 + 2ab = 8$. Следовательно, $(a + b)^2 + a^2 + b^2 = 8$.

Число 8 раскладывается в сумму трех квадратов единственным образом $8 = 0 + 4 + 4$, поэтому пара (a,b) может быть равна $(2, -2)$, $(-2, 2)$, $(2, 0)$, $(-2, 0)$, $(0, 2)$, $(0, -2)$. Учитывая, что $c = -a - b$, получим тройки $(2, -2, 0)$, $(-2, 2, 0)$, $(2, 0, -2)$, $(-2, 0, 2)$, $(0, 2, -2)$, $(0, -2, 2)$.

Ответ: $(4, 4, 3)$, $(-4, -4, -3)$, $(2, 2, 0)$, $(-2, -2, 0)$, $(1, 1, -3)$, $(-1, -1, 3)$, $(2, -2, 0)$, $(-2, 2, 0)$, $(2, 0, -2)$, $(-2, 0, 2)$, $(0, 2, -2)$, $(0, -2, 2)$.

Критерии оценивания

Рассмотрен случай $a = b$ — 5 баллов.

Рассмотрен случай $a \neq b$ — 15 баллов.

В случае $a \neq b$ доказано лишь, что $a + b + c = 0$ — 5 баллов.

За потерю каждого из ответов снимать 1 балл.

Задача 4.2.3.4. (20 баллов)

Тема: геометрия.

Условие

Пусть ABC треугольник, в котором O — центр описанной окружности и H — точка пересечения высот. Пусть AD, BE и FC — высоты треугольника ABC. Обозначим через A' точку, симметричную точке A относительно прямой EF. Докажите, что $HOA'D$ — вписанный четырехугольник.

Решение

Докажем, что A, O, A' лежат на одной прямой.

Пусть AO пересекает EF в точке P.

Заметим, что $\angle AOC$ — центральный, поэтому $\angle AOC = 2\angle B$. Кроме того, $AO = OC$, поэтому $\angle OAC = \angle OCA = (180^\circ - 2\angle B)/2 = 90^\circ - \angle B$.

Тогда
$$\angle OAC = 90^\circ - \angle B = \angle DAF.$$
Из вписанности четырехугольника $BCEF$
$$\angle AEP = \angle B.$$

Тогда треугольники PEA и DBA подобны и $\angle APE = \angle ADB = 90°$. Получаем

$$AO \perp EF,$$

и точки A, O, P, A' лежат на одной прямой.

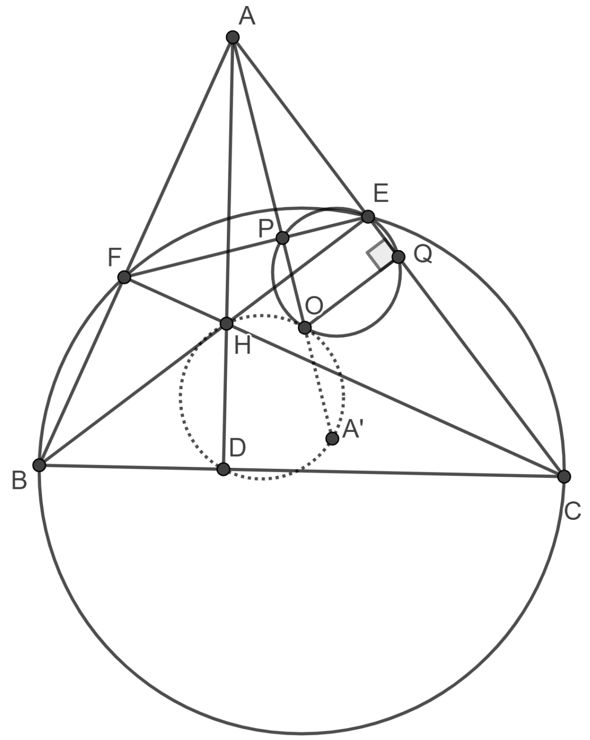


Рис. 4.2.7

Пусть Q середина AC. Тогда $OQEP$ вписанный.

Имеем

$$AO \cdot AP = AQ \cdot AE.$$

Умножим обе части на 2

$$AO \cdot (2AP) = (2AQ) \cdot AE.$$

Так как $2AP = AA'$ и $2AQ = AC$, то

$$AO \cdot AA' = AC \cdot AE = AD \cdot AH \quad \text{(так как } DHEC \text{ вписанный).}$$

Следовательно, $DHOA'$ — вписанный.

Критерии оценивания

Доказано, что $AO \perp EF$ — 5 баллов.

Доказано, что точки A, O, A' лежат на одной прямой — 5 баллов.

Задача 4.2.3.5. (25 баллов)

Темы: алгебра, функциональные уравнения.

Условие

Найдите все функции $f:\mathbb{R}\to\mathbb{R}$ такие, что

$$f(yf(x)-x)+f(xy-y)+f(x+y)=2xy$$

для всех $x,y\in\mathbb{R}$. $f:\mathbb{R}\to\mathbb{R}$ означает, что аргумент и значение функции f принадлежат множеству всех действительных чисел $\mathbb{R}$.

Решение

1. Подставим $x=y=0$. Получим, $3f(0)=0$, откуда $f(0)=0$.
2. Подставим $y=0$. Получим $f(-x)+f(x)=0$. Т. е. $f(-x)=-f(x)$ для всех $x\in\mathbb{R}$.
3. Подставим $y=-1$. Получим $f(-f(x)-x)+f(1-x)+f(x-1)=-2x$. Откуда $f(x+f(x))=2x$. Обозначим $k=-\frac{1}{2}+f(-\frac{1}{2})$. Тогда при $x=-1/2$ получим $f(k)=-1$.
4. Подставим вместо x число k, а вместо y подставим x. Получим $f(xf(k)-k)+$ $+f(kx-x)+f(k+x)=2kx$. Тогда $f(x(k-1))=2kx$.
   Если $k=1$, то $f(0)=2x$ и $0=2x$ для всех $x\in\mathbb{R}$. Противоречие.
   Если $k\neq 1$, то, обозначив, $z=x(k-1)$, получим $f(z)=\dfrac{2k}{k-1}z$. Поэтому $f(x)=cx$, для некоторого c.
   Подставим теперь в исходное уравнение $c(ycx-x)+c(xy-y)+c(x+y)=2xy$ или $cxy(c+1)=2xy$ для всех $x,y\in\mathbb{R}$. Откуда $c(c+1)=2$. Решая уравнение, находим $c=1$ или $c=-2$.

Ответ: $f(x)=x,\ f(x)=-2x$.

Критерии оценивания

Приведен один из ответов и проверено, что он подходит — 2 балла.

Приведены оба ответа и проверено, что они подходят — 5 баллов.

Баллы за указанные выше критерии не складываются.

Получено условие $f(-x)=-f(x)$ — 5 баллов.

Получено $f(x)=cx$ — 10 баллов.

Баллы за указанные продвижения складываются.

4.3. Инженерный тур

4.3.1. Общая информация

Задача заключительного этапа посвящена реидентификации диких животных (определению конкретных особей) по фото с фотоловушек.

Часть видов имеет индивидуальные особенности (окрас шерсти). Ученые создают каталоги с особями идентифицируемого вида животных и определяют каждую конкретную особь по фото, что повышает эффективность мониторинга вида.

Технологии искусственного интеллекта помогают быстрее и проще обрабатывать данные для реидентификации. Это экономит время зоологов, позволяя им больше сосредоточиться на спасении животных и научных исследованиях.

Участникам заключительного этапа необходимо решить задачу реидентификации амурских тигров по фото с фотоловушек с помощью технологии компьютерного зрения и metric learning. Эмбединги для фото одной и той же особи должны быть максимально близки между собой и как можно более отличаться от эмбедингов других особей.

4.3.2. Легенда задачи

Амурский тигр занесен в Красную книгу РФ. Мониторинг популяции помогает собрать данные о численности тигров, их распределении и состоянии среды обитания. Это является основой для разработки эффективных мер по защите и восстановлению популяции.

Фотоловушки предоставляют возможность отслеживать поведение тигров, их миграционные маршруты, предпочтения в выборе места обитания и взаимодействие с другими видами. Эти данные помогают исследовать экосистемные связи и влиять на сохранение всего вида на более широком уровне.

Ученые-зоологи определяют особей тигра по индивидуальному узору на шкуре животного на фото, при этом узор с каждой стороны различен. Реидентификация и мониторинг каждой отдельной особи позволяет более эффективно отслеживать популяцию тигров.

Необходимо обучить модель компьютерного зрения для формирования цифрового представления (эмбединга) для кропа с изображением тигра с фотоловушки.

4.3.3. Требования к команде и компетенциям участников

Количество участников в команде: 2 человека.

Компетенции, которыми должны обладать члены команды:

- Аналитик данных: поиск открытых данных и их исследовательский анализ.
- Data Scientist: обучение модели.

4.3.4. Оборудование и программное обеспечение

Таблица 4.3.1

Наименование	**Описание**
Ноутбуки: оперативная память — 16 Гб, хранилище SSD — 256 Гб, процессор — Intel Core i3.	Для доступа к Cloud, интернет ресурсам
Аккаунт на Cloud (параметры GPU V100 32Gb)	Обучение моделей

4.3.5. Описание задачи

Данные

Датасет представляет собой набор изображений в виде архива изображений (`train`, `test`) и файл `train.csv`. В нем присутствуют сложные примеры: фото, когда в кадр попала часть животного, его тело разделено препятствием, а также ночные и смазанные снимки. Данные были размечены специалистами по разметке данных совместно с учеными зоологами.

Сведения разделены на тренировочную и тестовую выборки в соотношении примерно 70% и 30% по особям.

Файл `train.csv` содержит следующие колонки:

- `file_name` — имя файла;
- `label` — название особи;
- `sequence` — номеру серии фотографий, к которой относится данное фото;
- `side` — сторона особи на фото (`left/right`).

Материалы для решения:

- папка с фотографиями для обучения (train): `https://disk.yandex.ru/d/t6YCFGkgodFOCA/train.zip`.
- папка с фотографиями для тестовой части (test): `https://disk.yandex.ru/d/t6YCFGkgodFOCA/test.zip`.
- файл с разметкой для обучающей части: `https://disk.yandex.ru/d/t6YCFGkgodFOCA/train.csv`.

Baseline (базовое решение)

Приведенный пример решения задачи реидентификации животных на изображении основан на библиотеке `open metric learning` (архитектура `resnet50`).

Бейзлайн позволяет пройти путь от установки библиотек и обучения модели до получения файла с предсказаниями и логов обучения.

Ссылка на бейзлайн: https://disk.yandex.ru/d/t6YCFGkgodFOCA/baseline.zip.

Формат решения

Соревнование подразумевает отправку файла с предсказаниями модели на платформу для расчета метрики.

Сам файл представляет собой `csv`-документ с двумя колонками:

- `image_name`;
- `recommendation`.

Строки — это пары с названием файла изображения и соответствующего отсортированному списку всех фото из тестовой выборки (начиная от самого похожего изображения до менее похожего).

Допускается использование открытых (доступных в сети) датасетов, позволяющих свободное некоммерческое использование.

Ручная разметка данных запрещена.

Все участники должны предоставить на проверку воспроизводимые решения.

4.3.6. Система оценивания

Участники отправляют `csv`-файл с предсказаниями с двумя колонками `image_name` (`query image`) и `recommendation*`.

Строки — это пары с названием файла изображения и соответствующего отсортированному списку всех фото из тестовой выборки (начиная от самого похожего изображения до менее похожего).

Схожесть изображений оценивается любой мерой близости векторов (на выбор участников).

Количество строк равно количеству изображений в тестовой выборке.

Для подсчета метрики из списка `recommendation` удаляются все изображения той же серии, что и `query image`, включая само `query image`.

Далее учитываем только одно (первое) вхождение каждой особи в список `recommendation`.

Для оценки качества работы модели реидентификации используется метрика, взвешенная CMC по особям тестового набора данных

$$cmc_w = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=0}^{4} w[i] * sign(find(j) == i),$$

где

$w = [1, 0{,}9, 0{,}8, 0{,}7, 0{,}6]$;

$sign(*)$ — индикаторная функция (1, если истина, иначе 0);

$find(j)$ — номер первого вхождения особи j, исключая эту же особь из той же серии, что и фотография запроса j.

- 1 — правильная особь на первом месте;
- 0,9 — правильная особь на втором месте;
- 0,8 — правильная особь на третьем месте;
- 0,7 — правильная особь на четвертом месте;
- 0,6 — правильная особь на пятом месте;
- 0 — правильная особь на шестом месте или дальше.

Расчет этого значения проводилась на платформе ODS: https://ods.ai/competitions/nto24-25-final.

4.3.7. Решение задачи

По ссылке представлено решение команды-победителя feel the asi, показавшее наилучший результат (private score: 0,9742990654): https://disk.yandex.ru/d/t6YCFGkgodFOCA/solution1.zip.

Структура архива с решением:

- `README.md` — описание запуска алгоритмов;
- `inferense.ipynb` — код для получения файла посылки;
- `requirements.txt` — список всех необходимых библиотек для запуска обучения и инференса;
- `train.ipynb` — файл для обучения модели;
- `train_without_val.ipynb` — файл для обучения модели без валидации;
- `models` — папка с весами модели после обучения;
- `top1_solution.csv` — файл посылки.

Решение основано на библиотеке OML (open-metric-learning). Версия Python 3.10.12.

Используемые библиотеки:

1. `conda create -n nto python=3.10`
2. `pip install -r requirements.txt`

Применяемые датасеты:

- предложенный датасет от организаторов;
- опционально ATRW.
  Подготовка датасета для обучения:
- использовать окружение с `requirements.txt`;
- запустить блок `Dataset create` в файле `train.ipynb`.

Для обучения каждой модели можно задать название `backbone image_size` и наличие в обучающей выборке ATRW, в названиях наших моделей указана версия `efficientnet` который является `backbone`, например `b0`, `b1` и т. д.

Число в названии модели обозначает `image_size`, а `___` в конце то, что модель обучалась кодом `train_without_val.ipynb`.

Обучение модели для Re-ID:

- использовать окружение с `requirements.txt`;
- запустить блок `Model` в файле `train.ipynb`.

Формирование файла для отправки на платформу:

- использовать окружение с `requirements.txt`;
- запустить `inferense.ipynb`.

4.3.8. Материалы для подготовки

1. Kevin Musgrave — PyTorch Metric Learning [Электронный ресурс]. — Режим доступа: https://kevinmusgrave.github.io/pytorch-metric-learning/.
2. TorchRec — A PyTorch Domain Library for Recommendation Systems [Электронный ресурс] // PyTorch Blog. — Режим доступа: https://pytorch.org/blog/introducing-torchrec/.
3. Zhou K. Deep Person Re-identification: MODEL ZOO [Электронный ресурс]. — Режим доступа: https://kaiyangzhou.github.io/deep-person-reid/MODEL_ZOO.
4. Open Metric Learning — OML-Team GitHub Repository [Электронный ресурс]. — Режим доступа: https://github.com/OML-Team/open-metric-learning.

5. Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи предметного тура по двум предметам: математике и информатике и инженерного тура. В каждом предмете максимально можно было набрать 100 баллов, в инженерном туре 100 баллов. Для того чтобы пройти во второй этап, участники должны были набрать в сумме по обоим предметам и инженерному туру не менее 50,0 баллов, независимо от уровня.

Второй отборочный этап

Количество баллов, набранных при решении всех задач второго отборочного этапа, суммируется. Победители второго отборочного этапа должны были набрать не менее 100,0 баллов, независимо от уровня.

Заключительный этап

Индивидуальный предметный тур

- математика — максимально возможный балл за все задачи — 100 баллов;
- информатика — максимально возможный балл за все задачи — 100 баллов.

Командный инженерный тур

Команды заключительного этапа получали за командный инженерный тур от 0 до 100,00 баллов: команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем.

Все результаты команд нормировались по формуле:

$$\frac{100 \times x}{MAX},$$

где x — число баллов, набранных командой,

MAX — число баллов, максимально возможное за инженерный тур.

В заключительном этапе олимпиады индивидуальные баллы участника складываются из двух частей, каждая из которых имеет собственный вес: баллы за индивидуальное решение задач по предмету 1 (математика) с весом $K_1 = 0{,}15$, по предмету 2 (информатика) с весом $K_2 = 0{,}15$, баллы за командное решение задач инженерного тура с весом $K_3 = 0{,}7$.

Итоговый балл определяется по формуле:

$$S = K_1 \cdot S_1 + K_2 \cdot S_2 + K_3 \cdot S_3,$$

где S_1 — балл первой части заключительного этапа по математике (предметный тур) ($S_{1\text{ макс}} = 100$);

S_2 — балл первой части заключительного этапа по информатике (предметный тур) ($S_{2\text{ макс}} = 100$);

S_3 — итоговый балл инженерного командного тура ($S_{3\text{ макс}} = 100$).

Итого максимально возможный индивидуальный балл участника заключительного этапа — 100 баллов.

Критерий определения победителей и призеров

Чтобы определить победителей и призеров (независимо от класса) на основе индивидуальных результатов участников, был сформирован общий рейтинг всех участников заключительного этапа. С начала рейтинга были выбраны 7 победителей и 16 призеров (первые 25% участников рейтинга становятся победителями или призерами, из них первые 8% становятся победителями, оставшиеся — призерами).

Критерий определения победителей и призеров (независимо от уровня)

Категория	**Количество баллов**
Победители	71,57 и выше
Призеры	От 58,12 до 71,57

6. Работа наставника после НТО

Участие школьника в Олимпиаде может завершиться после любого из этапов: первого или второго отборочных, либо после заключительного этапа. В каждом случае после завершения участия наставнику необходимо провести с учениками рефлексию — обсудить полученный опыт и проанализировать, что позволило достичь успеха, а что привело к неудаче. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: https://academy.sk.ru/events/310.

Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (P2P). Они могут выявить внутренние проблемы команды, для решения которых в план подготовки можно добавить мероприятия, направленные на ее сплочение.

Стоит рассказать, что в истории НТО было много примеров, когда не победив в первый раз, на следующий год участники показывали впечатляющие результаты, одержав победу сразу в нескольких профилях. Конечно, важно отметить, что так происходит только при учете прошлых ошибок и подготовке к Олимпиаде в течение года.

Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с ними помогает наставнику продемонстрировать важность компетенций, развиваемых в процессе участия в НТО, для будущего образования и карьеры школьников. Поддержка родителей помогает мотивировать участников и позволяет выделить необходимое время на занятия в кружке.

С участниками-выпускниками наставнику рекомендуется обсудить их дальнейшее профессиональное развитие и его связь с выбранными профилями НТО. Отдельно можно обратить внимание на льготы для победителей и призеров, предлагаемые в вузах с интересующими ученика направлениями. Кроме того, ряд вузов предлагает льготы для всех финалистов НТО, а также учитывает результаты Конкурса цифровых портфолио «Талант НТО».