МАТЕРИАЛЫ ЗАДАНИЙ

Всероссийской междисциплинарной олимпиады школьников

«Национальная технологическая олимпиада»

по профилю

«Искусственный интеллект»

2022/23 учебный год

http://ntcontest.ru

Оглавление

Введение

Национальная технологическая олимпиада

Всероссийская междисциплинарная олимпиада школьников «Национальная технологическая олимпиада» (далее — 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».

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.

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» — это командная инженерная Олимпиада, позволяющая школьникам работать в 39-ти 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.

Заявки на участие в Национальной технологической олимпиаде 2022/23 учебного года подали 135 тысяч участников из 85-ти регионов России и зарубежных стран. Общий охват соревнований с 2015 года превысил 350 тысяч человек.

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

- 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 и ведущими экспертами, а также найти единомышленников.

Ценности НТО

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Особое пространство Олимпиады создают общие ценности и смыслы, которые предлагается разделять всем: участникам, организаторам, наставникам, экспертам.

Основа всей олимпиады — это современное технологическое образование как новый уклад жизни в современном мире. Этот уклад подразумевает доступность качественного образования для каждого заинтересованного человека, возможность постепенно и непрерывно учиться и развиваться, совместно создавать среду, в которой гуманитарное знание и новые технологии взаимно дополняют друг друга. Это идеал будущего общества. Участники Олимпиады уже сейчас попадают в такое будущее.

Как организаторы мы надеемся, что принципы, заложенные в основу НТО, станут общими принципами для всех, кто имеет отношение к Олимпиаде.

Решать прикладные задачи, нацеленные на умножение общественного блага

В соревнованиях и подготовке к ним мы адаптируем реальные задачи современной науки и производства к знаниям и навыкам, которые могут освоить школьники и студенты. Задачи имеют прикладное значение для людей и не оторваны от реальности. Мы стремимся к тому, чтобы участники понимали, для чего нужно решать такие задачи, кому в мире станет лучше, если они будут решаться системно и профессионально. Ценность Олимпиады заключается в том, что здесь можно попробовать себя в этом, и найти единомышленников для решения подобных задач в будущем.

Создавать, а не только потреблять

Создание новых решений мы ставим выше стремления потреблять уже созданное. Создание ценности для других ставим выше поиска личной выгоды. Это не значит, что нужно забыть о себе и самоотверженно посвятить всю свою жизнь делу технологического прогресса. Но творчество всегда приносит большую радость, чем потребление. Это относится и ко всей олимпиаде.

Олимпиада — это общее дело организаторов, партнеров и участников. Способность принимать проблемы олимпиады как свои и пытаться решить их ценнее для творческого человека, чем желание найти недостатки в работе других.

Работать в команде

Способность работать в команде — это не только эффективная стратегия действия в современном мире. Работа в команде не отрицает наличия свободной воли каждого конкретного участника, его значимости и права на собственное мнение. Но в сообща мы стремимся достигнуть общей цели, опираясь на взаимное уважение всех участников, учитывая интересы и слабые и сильные стороны каждого.

Команды формируют целые сообщества, которые имеют сходные цели и ценности и могут очень многое, поскольку сильные горизонтальные связи помогают реализовывать самые дерзкие и амбициозные задачи. Это то, что нужно для технологического развития. Мы заняты построением такого сообщества и надеемся, что вы захотите стать его частью.

Осваивать и ответственно развивать новые технологии

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ. Это прежде всего сообщество людей, увлеченных современными технологиями. Нас всех объединяет стремление разобраться в них, создать что-то новое и найти таких же увлеченных единомышленников.

Мы — часть сообщества технологических энтузиастов, и для нас границы возможностей технологий всегда подвижны. Именно поэтому просим не забывать об этике инженера и ученого, ответственности за свои изобретения перед людьми, которых это касается. Творя новое, не навреди!

Играть честно и пробовать себя

Мы признаем, что победа в соревнованиях важна и нужна. Но утверждаем, что для победы не все средства хороши и цель не является оправданием для грязной игры. Победа должна быть заслужена в рамках правил, единых для всех. Человек, который играет честно, не будет списывать, интриговать, подставлять других и заниматься прочей нездоровой конкуренцией.

Человек, который играет честно, — уважает себя, свою команду и соперников. Он принимает правила игры и в заданных рамках доказывает право на победу.

Мы бережем пространство Олимпиады как безопасное для всех участников. Это помогает искать себя, и при этом не бояться пробовать новые задачи, определять свой дальнейший путь, учиться на ошибках и каждый год становиться более сильным и подготовленным.

Быть человеком

Соревнования — это очень сложный и эмоционально насыщенный процесс. Чтобы он приносил радость и пользу всем, мы призываем всех участников вести себя порядочно и думать не только о себе.

Вежливость, эмпатия и забота — вот что сделает процесс комфортным и полезным для всех. Мы ценим уважение труда каждого человека и его позиции, бережное отношение к работе и жизни каждого. И просим отказаться от токсичной оценочной критики — она не решит ваши проблемы, а сделает хуже вам, другому и всей Олимпиаде в целом.

Человек, который остается человеком, умеет признавать ошибки и отвечать за слова и дела перед другими. Здесь это ценят. Встав перед альтернативой между сиюминутной выгодой, капризом и общей целью соревнования — человек выберет последнее и поможет другим, организаторам и участникам, поддержать эту цель.

Важное замечание. Этот текст — живое выражение смыслов и ценностей Национальной технологической олимпиады. Он будет меняться вместе с развитием нашего сообщества. Авторы с благодарностью примут помощь от всех, кто чувствует сопричастность ценностям и готов включиться в их доработку.

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 направлениям:

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

В 2022/23 учебном году 28 профилей НТО включены в Перечень олимпиад школьников, 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, что 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 старшеклассников проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. На данном этапе участникам предлагаются задачи по двум предметам, соответствующим тому или иному профилю, а также задания, формирующие теоретические знания и представления по направлениям выбранных профилей.
- Второй отборочный этап — заочный командный. На данном этапе участникам предлагаются индивидуальные компетентностные и командные задачи, связанные с направлением выбранного профиля.
- Заключительный этап — очный командный. Этап представляет собой очные соревнования длительностью 5–6 дней, куда приезжают команды со всей страны, успешно справившиеся с двумя отборочными этапами, и решают комплексные прикладные инженерные задачи.

Профили НТО 2022/23 учебного года и соответствующий уровень РСОШ

Автоматизация бизнес-процессов	II

Профили НТО 2022/23 учебного года и соответствующий уровень РСОШ

Автономные транспортные системы	III
Анализ космических снимков и геопространственных данных	III
Аэрокосмические системы	III
Беспилотные авиационные системы	II
Большие данные и машинное обучение	III
Водные робототехнические системы	II
Геномное редактирование	III
Инженерные биологические системы. Агробиотехнологии	III
Интеллектуальные робототехнические системы	II
Интеллектуальные энергетические системы	III
Информационная безопасность	III
Искусственный интеллект	II
ИТ-Медицина	Новый профиль НТО
Композитные технологии	II
Летающая робототехника	III
Наносистемы и наноинженерия	III
Научная медиажурналистика	Не входит в перечень
Нейротехнологии и когнитивные науки	II
Новые материалы	III
Передовые производственные технологии	III
Разработка компьютерных игр	III
Робототехнические системы в исследованиях моря	Новый профиль НТО
Современная пищевая инженерия	Не входит в перечень
Спутниковые системы	III
Технологии беспроводной связи	II
Технологии виртуальной реальности	III
Технологии дополненной реальности	III
Технологическое мейкерство	Новый профиль НТО
Технологическое предпринимательство	Новый профиль НТО
Умный город	III
Урбанистика	Новый профиль НТО
Финансовый инжиниринг	III
Фотоника	Не входит в перечень
Цифровое производство в машиностроении	Новый профиль НТО
Цифровые сенсорные системы	Не входит в перечень
Цифровые технологии в архитектуре	III
Ядерные технологии	III

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

- учащихся технологических кружков, любители инженерных и робототехнических соревнований;
- олимпиадников, которым интересны межпредметные олимпиады;
- фанатов и адептов передовых технологий;
- школьников, участвующих в хакатонах, проектных конкурсах и школах;
- будущих предпринимателей, намеревающихся найти на Олимпиаде единомыш-

ленников для будущего стартапа;

- увлекающихся школьников, которые хотят видеть предмет шире учебника.

Познакомить школьников с НТО и ее направлениями, замотивировать принять участие в НТО можно с помощью специальных мероприятий: Урок НТО и Дни НТО. Как педагогу провести Урок НТО, или как в образовательном учреждении организовать День НТО можно познакомиться в методических рекомендациях на сайте НТО. Там же можно выбрать и скачать необходимые уроки и подборки материалов по направлениям https://nti-lesson.ru/.

Участвуя в НТО, школьники получают возможность работать с практикоориентированными задачами в области прорывных технологий, собирать команды единомышленников, включаться в профессиональное экспертное сообщество, а также заработать льготы для поступления в вузы.

У НТО есть площадки подготовки по всей стране, которые занимаются привлечением участников и проводят мероприятия по подготовке к соревнованиям. Они могут быть открыты:

- в организациях общего и дополнительного образования;
- на базе частных кружков в области программирования, робототехники и иных технологий;
- в вузах;
- технопарках

и других организациях.

Каждое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к олимпиаде, что дает возможность включиться в Кружковое движение НТИ.

На сайте НТО размещены инструкции о том, как организация может стать площадкой подготовки: https://ntcontest.ru/mentors/stat-ploshadkoi/. Условия регистрации и требования к работе площадок подготовки обновляются вместе с развитием олимпиады. Обновленная версия размещается на сайте перед началом нового цикла олимпиады.

Наставники НТО

В НТО большое внимание уделяется работе с наставниками. Наставник НТО оказывает всестороннюю поддержку участникам Олимпиады, помогая решать организационные вопросы и развивать как технические знания и компетенции, так и социальные навыки, связанные с работой в команде.

Наставником может стать любой человек, которому интересно сопровождать участников и помогать им формировать необходимые для решения технологических задач компетенции и готовиться к соревнованиям. Это может быть преподаватель школы или вуза, педагог дополнительного образования, руководитель кружка, эксперт в технологической области, представитель бизнеса и т. п. Если наставнику не хватает собственных знаний, он может привлекать коллег и внешних экспертов и поддерживать усилия и мотивацию учеников, которые разбирают задачи самостоятельно. На данный момент сообщество наставников НТО включает в себя более 7 тысяч человек.

Главная задача наставника — выстроить комплексную структуру подготовки к Олимпиаде в течение всего учебного года. В области ответственности наставника находится поддержка мотивации участников и помощь в решении возникающих проблем. Не менее важно зафиксировать цели и ожидания от предстоящих соревнований, что поможет оценить прирост профессиональных компетенций, личных и командных навыков за время подготовки.

Примеры организационных задач, которые стоят перед наставником НТО:

- Информирование. На этапе регистрации на Олимпиаду наставник привлекает участников, рассказывая, что такое НТО и какие преимущества она предлагает. Наставнику необходимо разобраться в устройстве НТО, этапах и расписании этапов, а также изучить профили, чтобы помочь каждому ученику выбрать наиболее перспективные и интересные для него направления.
- Формирование программы подготовки. Наставник составляет график подготовки к НТО и следит за его реализацией, руководя процессом подготовки учеников.
- Отслеживание сроков. Наставник следит за сроками проведения этапов НТО и напоминает участникам о необходимости своевременной загрузки решений на платформу.

Примеры задач наставника, связанных с непосредственной подготовкой к соревнованиям:

- Анализ компетенций участников. Наставник вместе с учениками оценивает компетенции, которые необходимы для успешного участия в НТО, выявляет нехватку знаний и навыков и отбирает материалы и задачи, которые ученикам нужно изучить и решить.
- Содержательная подготовка к первому и второму отборочному этапу. Наставник вместе с учениками изучает материалы для подготовки, рекомендованные разработчиками выбранных профилей, а также разбирает и решает задачи НТО прошлых сезонов. Рекомендуется использовать записи вебинаров, материалы и онлайн-курсы профилей.
- Содержательная подготовка к заключительному этапу. Наставник может использовать разборы задач заключительного этапа прошлых лет, а также следить за расписанием подготовительных очных и дистанционных мероприятий и рекомендовать ученикам их посещать.

Примеры задач наставника в области развития социальных навыков, связанных с развитием личной эффективности и взаимодействия с другими участниками:

- Формирование команд. Второй отборочный этап НТО проходит в командном формате. Наставник помогает ученикам сформировать эффективную команду с оптимальным распределением ролей. В ряде случаев он может содействовать в поиске недостающих участников команды, в том числе в других городах и стать наставником такой команды, коммуникация в которой осуществляется через web-сервисы.

- Отслеживание прогресса и анализ полученного опыта. Наставник проводит рефлексию прогресса отдельных участников и команды по результатам каждого этапа НТО и после завершения участия в соревнованиях. Это помогает участникам оценить свое движение по траектории соревнований, сильные и слабые стороны, сформулировать, каких компетенций не хватило для более высокого результата и как их можно улучшить в будущем.
- Поддержка и мотивирование участников. Наставник поддерживает интерес учеников к соревнованиям, а также помогает им сохранять высокую мотивацию, что особенно важно, если команда показала результаты хуже, чем ожидалось.
- Выстраивание индивидуальной образовательной траектории. Наставник может помочь ученикам осознанно создать собственную траекторию развития, в том числе вне НТО: подбор обучающих курсов и соревнований, выбор вуза и направления дальнейшего обучения.

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/promo. Курс формирует общие представления о работе наставников в области подготовки участников к инженерным соревнованиям.

Наставникам предлагаются образовательные программы, разработанные на основе восьмилетнего опыта организации подготовки к НТО. В настоящий момент такие программы представлены по 10-ти передовым технологическим направлениям:

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

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.

Формируется сообщество наставников НТО. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: https://konkurs.kruzhok.org, принять участие в котором может каждый наставник. По итогам конкурса кружки-участники размещаются на Всероссийской карте кружков: https://map.kruzhok.org.

В 2022 году был разработан Навигатор для наставников команд или отдельных участников НТО: https://www.notion.so/bdlv/5a1866975c2744728c2bd8ba80d21ec2.

Навигатор ориентирован на начинающих наставников и помогает погрузиться в работу с НТО. Опытным наставникам Навигатор может быть полезен как сборник важных рекомендаций и статей:

- Смогут ли мои ученики принять участие в НТО.
- Как наставнику зарегистрироваться в НТО.
- Как помочь участникам выбирать профили.
- Что можно успеть сделать, если я и мои ученики начнем участвовать с нового учебного года.
- Как убедить руководство включиться в НТО.
- Что важно знать, начиная подготовку школьников.
- Как организовать подготовку.
- Как проводить рефлексию.
- Как мотивировать участников.
- Как работать с командой участников НТО.

Организаторы Олимпиады также оказывают экспертно-методическую поддержку сообществу наставников. Были разработаны методические рекомендации для наставников: «Технологическая подготовка инженерных команд»: https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih. Рассмотрены особенности подготовки к 5-ти направлениям:

- Большие данные.
- Машинное обучение.

- Искусственный интеллект.
- Спутниковые системы.
- Летающая робототехника.

Искусственный интеллект

Цель профиля «Искусственный интеллект» — развитие у школьников прикладных навыков в сфере искусственного интеллекта. Профиль знакомит школьников с методами машинного и глубоко обучения для решения актуальных для науки и бизнеса задач.

Соревнования по профилю способствуют повышению уровня обеспечения российского рынка технологий искусственного интеллекта квалифицированными кадрами за счет повышения привлекательности конкурсов и олимпиад, направленных на развитие интеллектуальных и творческих способностей обучающихся, в соответствии с Национальной стратегией развития искусственного интеллекта Российской Федерации, а также вовлекает школьников в сферу искусственного интеллекта и ориентирует на профессиональное развитие в ней.

В 2022/23 учебном году участникам Олимпиады по профилю «Искусственный интеллект» было предложено погрузиться в задачу обработки видеоданных и соприкоснуться с обработкой естественного языка (NLP).

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

Первый отборочный (индивидуальный) дистанционный этап содержал задания предметных туров по математике и информатике, а также инженерный тур, в рамках которого участникам было предложено решить базовую задачу машинного обучения: используя предоставленные табличные данные с характеристиками космических объектов на космических снимках, участникам нужно было предсказать является ли наблюдаемая звезда карликом или гигантом. Благодаря решению данной задачи участники могли познакомиться с базовыми алгоритмами машинного обучения и соревновательной платформой.

Предметный тур определял уровень подготовки школьников по предметам: математика и информатика. Задачи по информатике относятся к разделам: алгоритмы, программирование и методы оптимизации. Здесь школьники должны были 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 проверяли у участников знания по алгебре, комбинаторике, теории вероятности и математической статистике. Таким образом, задачи предметного тура выявляли у участников знания, необходимые для решения задач следующего (второго) и заключительного этапов.

Дополнительно для участников был проведен онлайн-интенсив AI-ARROW. Участие в интенсиве подразумевало 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** участники работали с данными и видео. Им было нужно проанализировать короткие (до 1 мин.) видеоролики и распределить их по 10 заданным классам. Задача второго отборочного этапа решалась индивидуально. Задача позволила участникам понять принципы работы с видео форматом, а также, благодаря тому, что данная задача по своей сути представляла классическую задачу классификации, почувствовать разницу в использовании базовых алгоритмов машинного обучения и нейронных сетей.

В целях подготовки финалистов к заключительному этапу дополнительно был проведен образовательный хакатон, где финалисты получили возможность отточить необходимые для решения задачи заключительного этапа компетенции, а участники, не прошедшие в заключительный этап, — попробовать себя в решении задач более высокого уровня.

Заключительный этап НТО по профилю «Искусственный интеллект» состоял из командного инженерного и индивидуального предметного туров.

Предметный тур проводился по двум предметам: информатика и математика, и проверял предметные 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 индивидуальный результат участников.

Задача командного инженерного тура заключительного этапа предполагала создание алгоритма, способного ответить на вопросы к видео, заданные на естественном (русском) языке. Задачи автоматического анализа визуальных данных, в частности видео, позволяют решать самые разные задачи: быстрый поиск и фильтрация нужного контента, извлечение информации из данных и многое другое. Все это в значительной степени сокращает объем анализа по сравнению с ручной обработкой и создает возможности для автоматизации решения таких задач как: поиск нужного фрагмента видео по запросу, например, на камерах видеонаблюдения, поиск в базе данных различных видео, например, классификация видео с определенными участниками и другие.

По сравнению с предыдущими этапами, участники должны были продемонстрировать не только компетенции в области компьютерного зрения, но также компетенции в области обработки естественного языка (NLP).

Таким образом Учащиеся 8–11 классов, прошедшие все этапы НТО, демонстрируют понимание основных операций, необходимых для построения модели машинного обучения: подготовка и анализ данных с целью выявления признаков, необходимых для решения поставленной задачи, построение самой модели и интерпретация результатов ее работы для оценки качества.

Для того чтобы сделать это возможным в ходе олимпиады проводился цикл образовательных и отборочных мероприятий. С момента регистрации для подготовки участникам были доступны:

- материалы онлайн-буткемпа по искусственному интеллекту «AI-ARROW»: https://stepik.org/course/124175;
- материалы Академии искусственного интеллекта для школьников: https://ai-academy.ru/.

Работа наставника НТО на первом отборочном этапе

На первом отборочном этапе НТО участникам предлагаются задачи по предметам, соответствующим выбранным профилям. Для подготовки к первому отборочному этапу Олимпиады наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Разбор задач первого отборочного этапа НТО прошлых лет.
- Мини-соревнования по решению задач предметных олимпиад муниципального уровня.
- Углубленные занятия по разделам предметов в соответствии с рекомендациями разработчиков профилей.

Для проверки, самостоятельного решения или проведения мини-соревнований могут использоваться предметные курсы НТО на платформе Stepik. Также возможно привлечение других преподавателей-предметников для проведения занятий в случае, если у наставника недостаточно компетенций в области предметных олимпиад.

Первый отборочный этап

Предметный тур. Информатика

Первая попытка. Задачи 8–11 класса

Задача II.1.1.1. Три квадрата (15 баллов)

Темы: математика, задачи для начинающих.

Условие

Фермер владеет участком земли в форме прямоугольника с длинами сторон a и b. Недавно фермер понял, что может разбить свой участок на три части так, что каждая часть будет иметь форму квадрата, и решил воспользоваться этой возможностью. Напишите программу, которая найдет площадь каждой части после разбиения.

Формат входных данных

На вход подается два натуральных числа a и b — длины сторон прямоугольника. Числа не превосходят 1000. Каждое число подается в отдельной строке. Гарантируется, что длины сторон таковы, что прямоугольник может быть разбит на три квадрата.

Формат выходных данных

Требуется вывести через пробел три натуральных числа — площади каждого из участков после разбиения. Числа могут выводиться в произвольном порядке.

Методика проверки

Программа проверяется на 12-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

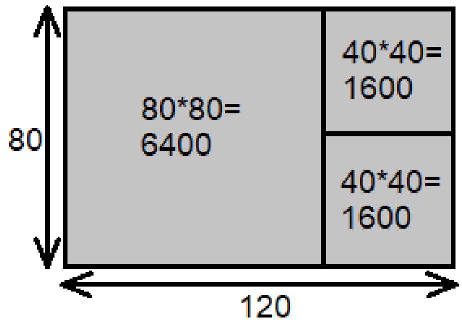
Примеры

Пример №1

Стандартный ввод
120 80
Стандартный вывод
6400 1600 1600

Пояснения к примеру

При заданных размерах прямоугольник может быть разбит на три квадрата так, как показано на рисунке ниже. Обратите внимание, что могут существовать и другие варианты разбиения.



Пример программы-решения

Ниже представлено решение на языке Python 3.

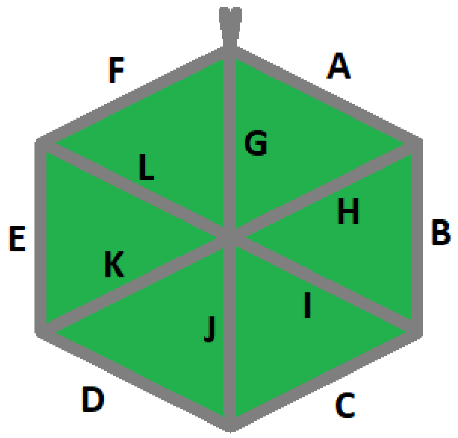
```
a, b = map(int,input())
if a>b:
    a,b=b,a
s1=a*a
s2=(a*b-s1)//2
print(s1,s2,s2)
```

Задача II.1.1.2. Пробежка в шестиугольном парке (18 баллов)

Темы: конструктивное построение, задачи для начинающих.

Условие

Иван Иванович совершает пробежки по парку, который имеет форму шестиугольника. В парке 12 аллей, обозначенных символами латинского алфавита от «A» до «L». Схему парка смотрите на рисунке. Длина каждой аллеи ровно 100 м. В парке есть только один вход у перекрестка аллей «A», «F», «G». Иван Иванович хочет начать и закончить пробежку у входа в парк и пробежать ровно k м. На каждом перекрестке Иван Иванович может повернуть в любую строну, но он не хочет поворачивать назад.



Напишите программу, которая составит любой маршрут движения, удовлетворяющий указанным требованиям.

Формат входных данных

На вход в подается одно натуральное число k — желаемая длина маршрута, $300 \leqslant k \leqslant 10000$. Число k делится на 100 без остатка.

Формат выходных данных

Требуется вывести строку из $k/100$ символов, содержащую обозначения аллей в построенном маршруте.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
800

Стандартный вывод
FLKEFAHG

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
a=int(input())//100
ans=''
while a>=6:
    ans+='AHG'
```

```
    a-=3
sol=['','','','AHG','ABIG','ABCJG']
print(ans+sol[a])
```

Задача II.1.1.3. Знакопеременная сумма (25 баллов)

Темы: *префиксные суммы.*

Условие

Знакопеременной суммой последовательности чисел $a_1, a_2, \ldots, a_k$ называется результат вычисления выражения $a_1 - a_2 + a_3 - a_4 + a_5 - \ldots + (-1)^{k+1}a_k$. Другими словами, мы складываем все числа в последовательности, но элементы с нечетными номерами мы берем со знаком плюс, а четные со знаком минус.

Задан массив $x_1, x_2, \ldots, x_n$ из n целых чисел и m запросов. Каждый запрос содержит по два натуральных числа b и e. В ответ на каждый запрос ваша программа должна взять подмассив с номерами элементов от b до e включительно и посчитать его знакопеременную сумму $x_b - x_{b+1} + x_{b+2} - x_{b+3} + \ldots + (-1)^{e-b}x_e$

Формат входных данных

На вход в первой строке подается одно натуральное число n размер массива чисел, $1 \leqslant n \leqslant 10^5$. Во второй строке через пробел записаны элемент массива целые числа $x_1, x_2, \ldots x_n$. Каждое из чисел не превосходит 10^6 по абсолютной величине. Далее в третьей строке записано одно натуральное число m количество запросов, $1 \leqslant m \leqslant 10^5$. В каждой из m последующих строк записано по два числа b_i и e_i таких, что $1 \leqslant b_i \leqslant e_i \leqslant n$. Каждая пара чисел задает границы подмассива для выполнения одного запроса.

Формат выходных данных

Требуется вывести через пробел m целых чисел $s_1, \ldots s_m$. Каждое из чисел должно быть равно знакопеременной сумме соответствующего подмассива.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a,end=' ')`.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых 10-ти тестах размер массива и количество запросов не превосходят 1000.

Примеры

Пример №1

Стандартный ввод
5
7 -4 1 3 2
4
2 2
1 3
2 4
4 5

Стандартный вывод
-4 12 -2 1

Пояснения к примеру

Для каждого из запросов ответ получается следующим образом:

$$-4 = -4$$

$$7 - (-4) + 1 = 12$$

$$-4 - 1 + 3 = -2$$

$$3 - 2 = 1$$

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n=int(input())
X=list(map(int,input().split()))
S=[0]
m=1
for x in X:
    S.append(S[-1]+m*x)
    m=-m
m=int(input())
ans=''
for i in range(3,m+3):
    b,e=map(int,input().split())
    s=S[e]-S[b-1]
    if b%2==0:
        s=-s
    print(s,end=' ')
```

Задача II.1.1.4. Проверка корректности маршрута (25 баллов)

Темы: <u>реализация</u>.

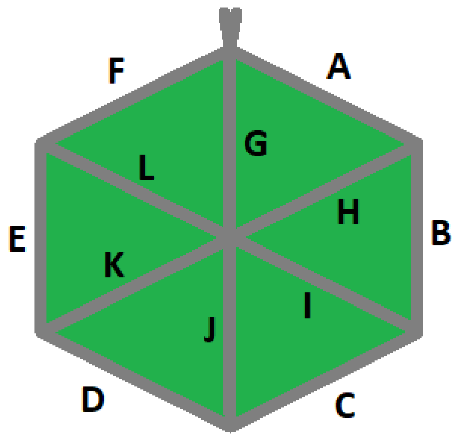
Условие

В некоторых случаях проверить корректность работы программы гораздо сложнее, чем написать ее. Сейчас у вас будет шанс в этом убедиться. От вас требуется написать программу для проверки корректности ответа второй задачи этой попытки. Напомним ее условие.

Имеется парк в виде шестиугольника с 12-ю аллеями, которые обозначены заглавными символами латиницы. В парке есть только один вход у перекрестка аллей «A», «F», «G». Схема парка приведена ниже. Требуется проверить корректность составленного маршрута движения по этому парку. Маршрут представляется как последовательность символов, представляющих аллеи в том порядке, в котором они были пройдены. Маршрут движения считается корректным если выполняются следующие требования.

- Описание маршрута содержит только символы от «A» до «L».
- Маршрут начинается и заканчивается у входа в парк.
- Запрещено разворачиваться на 180°. В частности, это означает, что начав движение с одного конца аллеи, вы обязательно дойдете до другого ее конца, причем на перекрестке вы должны будете перейти на другую аллею.

На вход вашей программе будет подано несколько описаний маршрутов. Ваша программа должна будет определить, какие из них удовлетворяют указанным требованиям.



Формат входных данных

На вход в первой строке подается одно натуральное число n — количество проверяемых маршрутов, $1 \leqslant n \leqslant 20$. Далее в n строках записаны сами маршруты. Описание каждого маршрута состоит из последовательности заглавных символов латиницы. Каждое описание не пустое, и содержит не более 100 символов.

Формат выходных данных

Программа должна вывести строку из n нулей и единиц. Единица на i-той позиции означает, что маршрут с номером i является корректным. В противном случае в этой позиции должен быть записан ноль.

Методика проверки

Программа проверяется на 5-ти тестах. Прохождение каждого теста оценивается в 5 баллов. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
6 ABCDKHA FMG ABBA ABCEF BCDEF ABCDK

Стандартный вывод
100000

Пояснения к примеру

Первый маршрут является корректным.

Второй маршрут содержит недопустимое обозначение аллеи.

В третьем маршруте происходит разворот на 180°.

Четвертый маршрут не является связным. После третьего шага он приходит к перекрестку «C», «D», «J» и с него нельзя попасть на аллею «E».

Пятый маршрут начинается не у входа.

Шестой маршрут заканчивается не у входа.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
def checkway(way):
    c=0
    p='Z'
    for x in way:
        if x<'A' or x>'L' or x==p:
            return False
        p=x
        num=ord(x)-ord('A')
        if num>5:
            if c==6:
                c=num-6
            elif c==num-6:
                c=6
            else:
                return False
        else:
```

```
            if c==num:
                c=(c+1)%6
            elif c==(num+1)%6:
                c=(c+5)%6
            else:
                return False
    return c==0

m=int(input())
for i in range(m):
    print(int(checkway(input())))
```

Вторая попытка. Задачи 8–11 класса

Задача II.1.2.1. Стрелки часов (12 баллов)

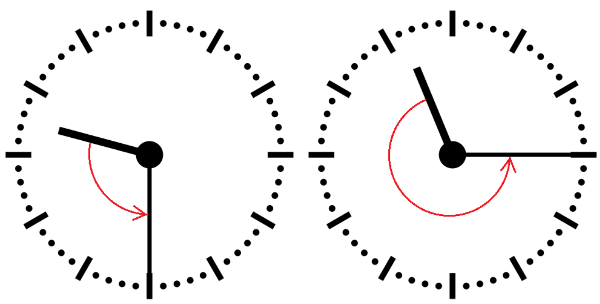
Темы: математика, задачи для начинающих.

Условие

Иван Иванович посмотрел на часы и заметил, что минутная и часовая стрелки образуют угол в $\alpha°$. С этого момента прошло h ч и m мин. Напишите программу, которая вычислит угол между стрелками после истечения этого времени.

Угол измеряется от часовой до минутной стрелки в направлении против хода часов. Если стрелки совпадают, то угол равен нулю. Рассмотрим пример на рисунке. Пусть $\alpha = 105°$. В частности, такой угол появляется в 9:30, так как минутная стрелка в этот момент указывает на $270°$, а часовая на $165°$. Через 1 ч 45 мин на часах будет 11:15. В этот момент времени минутная стрелка указывает на $0°$, а часовая на $112,5°$. Угол от часовой до минутной стрелки будет равен $360° - 112,5° = 247,5°$.

Отметим, что угол $\alpha = 105°$ появляется и в другие моменты времени, однако, это не повлияет на итоговый ответ.



Формат входных данных

На вход в первой строке подается одно целое неотрицательное число α исходный угол между стрелками, $0 \leqslant \alpha \leqslant 359$. Во второй строке через пробел на вход подается два числа h и m время, прошедшее с момента наблюдения в часах и минутах, $0 \leqslant h \leqslant 11$; $0 \leqslant m \leqslant 59$.

Формат выходных данных

Программа должна вывести одно вещественное число ответ к задаче. Ответ должен быть записан без погрешности.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
105 1 45
Стандартный вывод
247.5

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
a = float(input())
h,m = map(int,input().split())
a = a - 5.5*(m+h*60)
while a<0:
    a += 360
print(a)
```

Задача II.1.2.2. Номера домов (18 баллов)

Темы: <u>реализация.</u>

Условие

В поселке Березовом на улице Березовой построено n домов с номерами от 1 до n, причем дома с нечетными номерами расположены на одной стороне улицы, а с четными — на другой.

Рано утром дворник вышел к одному из крайних домов на этой улице, который имел номер k, и начал подметать тротуар, двигаясь от одного края улицы к другому краю. Потом он перешел на противоположную сторону улицы и начал подметать тротуар там, двигаясь назад.

Напишите программу, которая выведет номера домов, мимо которых проходил дворник, по известным числам n и k. Для лучшего понимания прочитайте пояснения к примерам.

Формат входных данных

На вход в одной строке подается два натуральных числа n и k — количество домов и номер одного из крайних домов на улице, $4 \leqslant n \leqslant 100$. Число k может принимать одно из четырех значений: $1, 2, n-1, n$.

Формат выходных данных

Ваша программа должна вывести через пробел последовательность номеров домов, в том порядке, в котором их проходил дворник.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a,end=' ')`.

Методика проверки

Программа проверяется на 36-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
`7 6`
Стандартный вывод
`6 4 2 1 3 5 7`

Пример №2

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

Пояснения к примеру

Рассмотрим первый пример. На улице 7 домов, дворник вышел к дому номер 6. Это означает, что он находится на четной стороне в конце улицы, и далее он будет двигаться к ее началу, проходя мимо домов 6, 4, 2. Потом он перейдет на противоположную сторону к дому номер 1 и пойдет к концу улицы, проходя мимо домов 1, 3, 5, 7.

Во втором примере на улице 8 домов, дворник вышел к дому номер 1. Это означает, что он находится на нечетной стороне в начале улицы, и далее он будет двигаться к ее концу, проходя мимо домов 1, 3, 5, 7. Потом он перейдет на противоположную сторону к дому номер 8 и пойдет к началу улицы, проходя мимо домов 8, 6, 4, 2.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
n,k=map(int,input().split())
if k==1:
    print(*range(1,n+1,2),*range(n-(n%2),0,-2))
elif k==2:
    print(*range(2,n+1,2),*range(n+(n%2)-1,0,-2))
elif k%2==1:
    print(*range(n+(n%2)-1,0,-2),*range(2,n+1,2))
else:
    print(*range(n-(n%2),0,-2),*range(1,n+1,2))
```

Задача II.1.2.3. Упорядочивание монет (25 баллов)

Темы: реализация, сортировки, теория графов.

Условие

В древнем кладе было найдено n монет различного веса. Каждая из монет была обозначена строчной буквой латиницы. Все обозначения были различными. Монеты были попарно взвешены на чашечных весах. Протокол взвешиваний состоял из $n(n-1)/2$ строк, каждая строка содержала ровно три символа. Первый и третий символ содержали обозначения монет, а во втором был записан результат сравнения: знак $<$ или знак $>$. Например, запись $d > b$ означает, что монета d тяжелее монеты b.

Взвешивания очень утомили лаборанта, и он просит вас написать программу, которая упорядочит монеты по возрастанию веса.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество монет, $4 \leqslant n \leqslant 26$. Далее в $n(n-1)/2$ строках записан протокол взвешиваний. Гарантируется, что протокол является корректным.

Формат выходных данных

Ваша программа должна вывести одну строку из n символов. Строка должна содержать обозначения монет в порядке возрастания их веса.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
4
b<x
k>b
x<k
b<d
x>d
d<k

Стандартный вывод
bdxk

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n=int(input())
cmp=set()
for i in range(n*(n-1)//2):
    cmp.add(input())
lst=list({s[0] for s in cmp} | {s[2] for s in cmp})
for i in range(len(lst)-1):
    for j in range(i+1,len(lst)):
        if lst[j]+'<'+lst[i] in cmp or lst[i]+'>'+lst[j] in cmp:
            lst[i],lst[j]=lst[j],lst[i]
print(''.join(lst))
```

Задача II.1.2.4. 2–3 дерево (25 баллов)

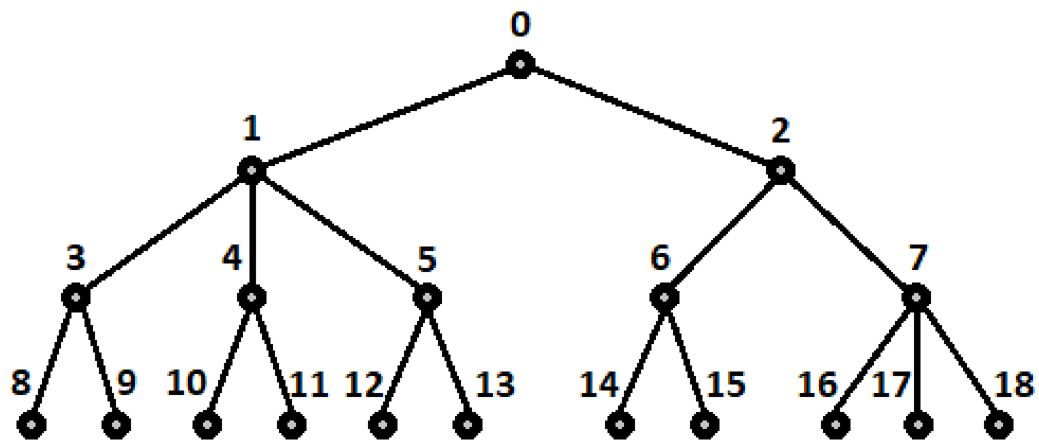
Темы: математика, графы и деревья, реализация.

Условие

Упорядоченное корневое дерево называется 2–3 деревом, если для него выполняются следующие условия:

- все узлы, кроме листьев, имеют два или три потомка;
- все листья находятся на одной высоте.

Пример 2–3 дерева приведен на рисунке ниже. Вы должны будете написать программу, которая составит произвольное 2–3 дерево с заданным количеством узлов или определит, что таких деревьев не существует.



Формат входных данных

На вход в первой строке подается одно натуральное число n — количество узлов в дереве, $2 \leqslant n \leqslant 100000$.

Формат выходных данных

Ваша программа должна вывести описание полученного дерева. Узлы дерева должны иметь номера от 0 до $n-1$ и быть упорядоченными сверху вниз и слева направо, как на рисунке. Для каждого узла, кроме корня, требуется указать номер его непосредственного предка. Вывод состоит из последовательности чисел $a_1, a_2, \ldots a_{n-1}$, записанных через пробел, где a_i это непосредственный предок узла с номером i.

Можно дать и другую интерпретацию. Вывод состоит из последовательности номеров нелистовых узлов, упорядоченных по возрастанию, причем номер каждого узла повторяется столько раз, сколько у него непосредственных потомков.

Если 2–3 дерево с указанным количеством узлов построить невозможно, то требуется вывести -1.

Методика проверки

Программа проверяется на 50-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
19

Стандартный вывод
0 0 1 1 1 2 2 3 3 4 4 5 5 6 6 7 7 7

Пример №2

Стандартный ввод
5
Стандартный вывод
-1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n=int(input())
k=1
curc=1
maxc=1
cnt=[1]
while maxc<n:
    maxc+=3**k
    curc+=2**k
    cnt.append(2**k)
    k+=1
if curc>n:
    print('-1')
else:
    for i in range(1,k):
        t=min(3*cnt[i-1]-cnt[i],(n-curc)//(2**(k-i)-1))
        curc+=t*(2**(k-i)-1)
        for j in range(i,k):
            cnt[j]+=t*(2**(j-i))
    m=0
    for i in range(k-1):
        p=cnt[i]
        s=cnt[i+1]
        while p>0:
            if s>2*p:
                print(m,m,m,end=' ')
                s-=3
            else:
                print(m,m,end=' ')
                s-=2
            p-=1
            m+=1
```

Третья попытка. Задачи 8–11 класса

Задача II.1.3.1. Количество нечетных чисел (12 баллов)

Темы: математика, задачи для начинающих.

Условие

Задан интервал целых чисел $[b; e]$. Вы должны написать программу, чтобы определить, сколько нечетных чисел принадлежит этому интервалу.

Обратите внимание, что интервал может быть достаточно большим, и решения, перебирающие все натуральные числа, не будут проходить часть тестов.

Формат входных данных

На вход в одной строке подается два целых числа b и e — границы интервала, $-10^{18} \leqslant b \leqslant e \leqslant 10^{18}$.

Формат выходных данных

Программа должна вывести одно число — количество нечетных чисел в заданном интервале.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
`-3 8`
Стандартный вывод
`6`

Пример №2

Стандартный ввод
`0 0`
Стандартный вывод
`0`

Пояснения к примеру

В первом примере указанному интервалу принадлежат шесть нечетных чисел: -3, -1, 1, 3, 5, 7.

Во втором примере интервал не содержит нечетных чисел.

Пример программы-решения

Ниже представлено решение на языке Python 3.

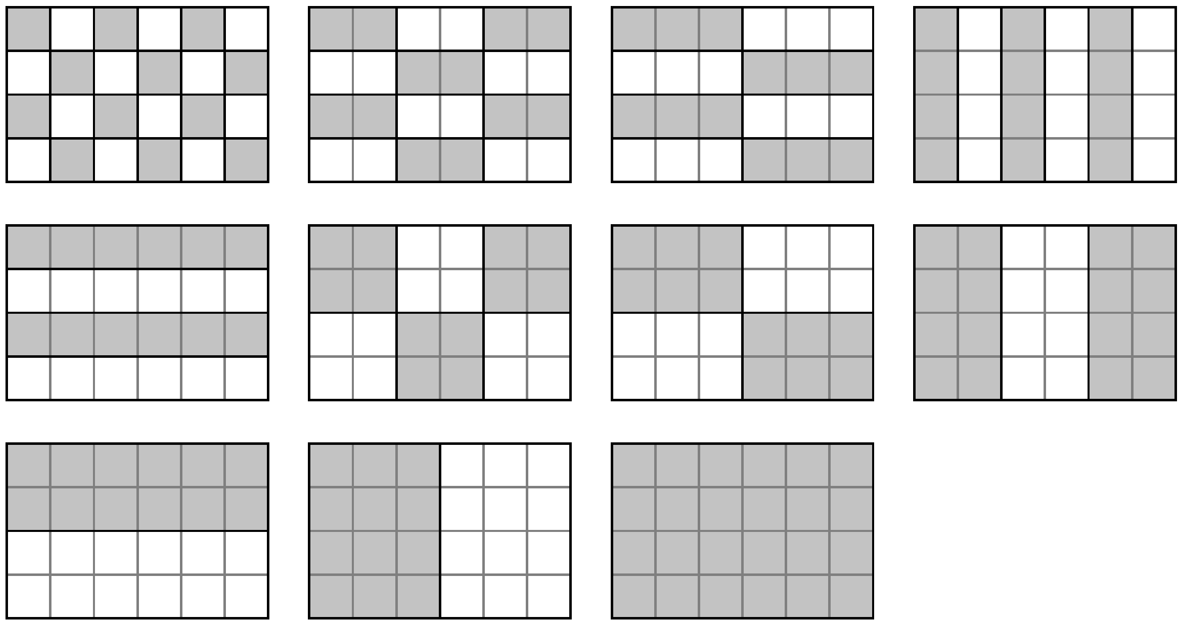
```python
b, e = map(int,input().split())
print(((e+e%2)-(b-b%2))//2)
```

Задача II.1.3.2. Наибольший общий делитель прямоугольников (18 баллов)

Темы: математика, задачи для начинающих.

Условие

Будем говорить, что прямоугольник P является делителем прямоугольника Q, если прямоугольник Q можно замостить прямоугольниками P, причем все они должны иметь одинаковую ориентацию. Например, делителями прямоугольника 6×4 будут следующие прямоугольники: 1×1, 2×1, 3×1, 4×1, 6×1, 2×2, 3×2, 4×2, 6×2, 4×3, 6×4. Примеры замощений можно увидеть на рисунке ниже. Обратите внимание, что прямоугольники $a \times b$ и $b \times a$ считаются одинаковыми.



Ваша задача заключается в написании программы, которая найдет наибольший общий делитель двух заданных прямоугольников. Из двух прямоугольников большим считается тот, площадь которого больше. Наибольших общих делителей может быть два. В этом случае допускается вывести любой из этих двух прямоугольников.

Формат входных данных

На вход в двух строках подаются размеры двух прямоугольников. Каждая строка содержит два натуральных числа — длину и ширину прямоугольника. Каждое из чисел не превосходит 10^{18}. Гарантируется, что введенные значения будут таковы, что площадь прямоугольника, который должен получиться в качестве ответа, не превысит 10^{18}.

Формат выходных данных

Программа должна вывести через пробел два числа — размеры искомого прямоугольника. Числа можно выводить в любом порядке.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
6 4 4 3

Стандартный вывод
4 3

Пример №2

Стандартный ввод
9 10 15 3

Стандартный вывод
3 5

Пример №3

Стандартный ввод
3 7 7 3

Стандартный вывод
3 7

Пример №4

Стандартный ввод
3 7 4 8

Стандартный вывод
1 1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
def gcd(a,b):
    while a>0 and b>0:
        if a>b:
            a%=b
        else:
```

```
            b%=a
    return a+b

a1,b1=map(int,input().split())
a2,b2=map(int,input().split())
a3=gcd(a1,a2)
b3=gcd(b1,b2)
a4=gcd(a1,b2)
b4=gcd(b1,a2)
if a3*b3>a4*b4:
    print(a3,b3)
else:
    print(a4,b4)
```

Задача II.1.3.3. Справедливый дележ (25 баллов)

Темы: реализация.

Условие

Два купца, живущие в разных городах, в далеком плавании купили несколько видов пряностей, и теперь хотят поделить их. Каждый из купцов будет продавать пряности только в своем городе, и цена каждой пряности в этих городах может отличаться. Купцы сочли, что будет справедливым, если они поделят пряности на две доли так, чтобы суммарная стоимость пряностей первой доли в первом городе была равна суммарной стоимости пряностей второй доли во втором городе. Существует несколько способов дележа, удовлетворяющих этому условию, но купцы хотят выбрать из них такой, при котором они получат максимум денег. Пряности являются сыпучим товаром, поэтому они могут быть поделены в любой пропорции

Рассмотрим пример. Есть три вида пряностей: перец, ваниль и корица. Стоимость всей партии перца в первом и втором городах составляет 120 и 200 условных единиц соответственно. Аналогичная стоимость партии ванили равна 180 и 140 условных единиц, а корицы — 100 и 60 условных единиц. Допустимым способом дележа будет, например, следующий: первый купец возьмет всю ваниль, второй — весь перец, а корицу они поделят поровну. Тогда стоимость доли первого купца в первом городе будет равна $180 + 100 \cdot 0,5 = 230$. Стоимость доли второго купца во втором городе составит $200 + 60 \cdot 0,5 = 230$. Стоимости долей равны, поэтому такой вариант дележа допустим. Но более выгодным будет другой вариант. Первый купец возьмет всю корицу и 3/4 ванили, а второй купец — весь перец и 1/4 ванили. Тогда стоимость доли в первом городе составит $100 + 180 \cdot 0,75 = 235$ и $200 + 140 \cdot 0,25 = 235$ во втором городе. Таким образом, второй вариант является более предпочтительным.

Напишите программу, которая найдет максимальную стоимость долей, при условии того, что дележ будет справедливым.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество видов пряностей, $1 \leqslant n \leqslant 100$. Во второй строке через пробел записаны n натуральных чисел $a_1, a_2, \ldots, a_n$ — цены всех видов пряностей в первом городе. Аналогично в третьей строке записаны числа $b_1, b_2, \ldots, b_n$ — цены всех видов пряностей во втором городе, $1 \leqslant a_i, b_i \leqslant 10^6$.

Формат выходных данных

Программа должна вывести одно число — максимальную стоимость долей. Это число может быть вещественным. Ответ будет считаться верным, если он отличается от ответа жюри не более чем на $0,01$.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пяти тестах $n \leqslant 3$. В первых 15 тестах $n \leqslant 10$. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
3 120 180 100 200 140 60
Стандартный вывод
235.0

Пример №2

Стандартный ввод
1 100 200
Стандартный вывод
66.66666666666667

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n=int(input())
A=map(float,input().split())
B=map(float,input().split())
p=list(zip(A,B))
p.sort(key=lambda x:x[1]/x[0])
s1,s2=0,0
i,j=0,n-1
while i<=j:
    if s1<s2:
        s1+=p[i][0]
        i+=1
    else:
        s2+=p[j][1]
        j-=1
if s1<s2:
```

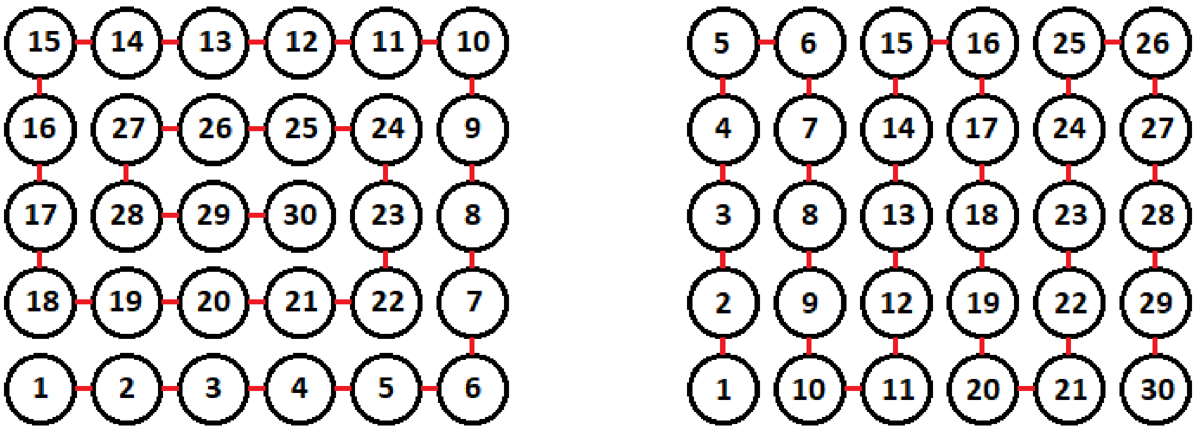
```
        s1+=(s2-s1)*p[j+1][0]/(p[j+1][0]+p[j+1][1])
    else:
        s1-=(s1-s2)*p[i-1][0]/(p[i-1][0]+p[i-1][1])
    print(s1)
```

Задача II.1.3.4. Чай с лимоном и сахаром (25 баллов)

Темы: <u>реализация.</u>

Условие

На прямоугольном столе в n рядов расставлены стаканы с чаем, в каждом ряде по m стаканов. Аня ходит вокруг стола и бросает в каждый стакан по ломтику лимона. Нумерация стаканов на рисунке слева соответствует той последовательности, в которой Аня переходит от одного стакана к другому. Яна ходит вдоль одного края стола туда и обратно, и бросает в каждый стакан кусочек сахара. Нумерация стаканов на рисунке справа соответствует той последовательности, в которой Яна переходит от одного стакана к другому.



Будем считать, что ломтик лимона и кусочек сахара в один стакан девочки бросают ровно за одну секунду. Напишите программу, которая найдет количество стаканов, в которых через t с лежит и лимон и сахар. В каждом тесте ваша программа должна будет ответить на k запросов. При этом количество и расположение стаканов на столе единое для всех запросов в одном тесте.

Формат входных данных

На вход в первой строке подается два натуральных числа n, m и k — количество рядов на столе, количество кружек в каждом ряду и количество запросов, $1 \leqslant n, m \leqslant 1000$, $1 \leqslant k \leqslant 10^5$. Во второй строке через пробел записано k натуральных чисел $t_1, t_2, \ldots, t_k$ — моменты времени, для которых требуется решить задачу, $1 \leqslant t_i \leqslant nm$. Каждый момент времени может встречаться более 1 раза.

Формат выходных данных

Программа должна вывести в одной строке через пробел k чисел — ответы для каждого из заданных моментов времени.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пятнадцати тестах $n, m \leqslant 10$. Тест из условия задачи при проверке не используется.

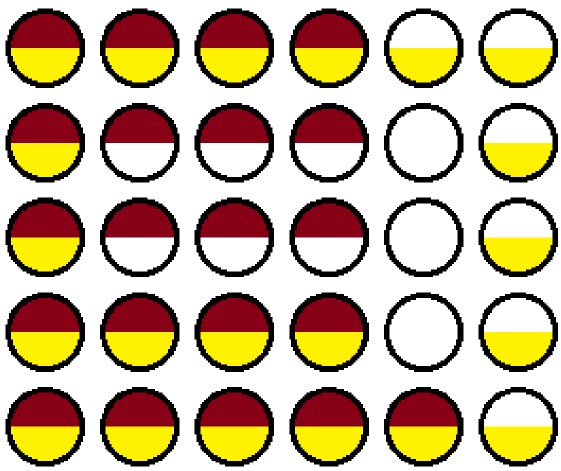
Примеры

Пример №1

Стандартный ввод
5 6 4 1 21 30 21
Стандартный вывод
1 15 30 15

Пояснения к примеру

На рисунке ниже показано решение задачи для теста из условия задачи после двадцать первой секунды. Желтым цветом помечены кружки с лимоном, коричневым — кружки с сахаром. Из рисунка видно, что в 15 чашках есть и лимон, и сахар.



Пример программы-решения

Ниже представлено решение на языке Python 3.

```
n,m,k=map(int,input().split())
F=[[0 for i in range(m)] for j in range(n)]
A=[0]
i1,j1,i2,j2,s=0,0,0,0,0
dr=0
for i in range(n*m):
    F[i1][j1]+=1
    F[i2][j2]+=2
    if F[i1][j1]==3:
```

```
        s+=1
    if (i2!=i1 or j2!=j1) and F[i2][j2]==3:
        s+=1
    A.append(s)
    if dr==0:
        if j1+1==m or (F[i1][j1+1]&1)==1:
            dr=1
    elif dr==1:
        if i1+1==n or (F[i1+1][j1]&1)==1:
            dr=2
    elif dr==2:
        if j1==0 or  (F[i1][j1-1]&1)==1:
            dr=3
    else:
        if i1==0 or (F[i1-1][j1]&1)==1:
            dr=0
    if dr==0:
        j1+=1
    elif dr==1:
        i1+=1
    elif dr==2:
        j1-=1
    else:
        i1-=1
    if j2%2==0:
        if i2==n-1:
            j2+=1
        else:
            i2+=1
    else:
        if i2==0:
            j2+=1
        else:
            i2-=1
for i in input().split():
    print(A[int(i)])
```

Четвертая попытка. Задачи 8–11 класса

Задача II.1.4.1. Сумма элементов списка (12 баллов)

Темы: математика, задачи для начинающих.

Условие

Алиса изучает списки в языке Python. По заданию из учебника она написала такую программу.

```
n = int(input())
x = [i%10 for i in range(n)]
print(sum(x))
```

Эта программа читает с консоли натуральное число n и делает список этой длины, состоящий из чисел от нуля до девяти, которые идут по кругу. Например, для $n = 25$ список будет иметь вид:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4]
```

В последней строчке на экран выводится сумма элементов этого списка. Для указанного списка, в частности, будет выведено число 100.

Боб очень недоволен учебником. Он считает, что учебник упускает главное — списки нужны для хранения информации, значимой для работы программы, а это задание можно выполнить, как минимум, без списков, а в идеале — без циклов и условий.

Напишите программу для этого задания, которую Боб сочтет удовлетворительной. Для этого она должна быстро и корректно работать для чисел до 10^{15}.

Формат входных данных

На вход подается одно натуральное число n, которое не превосходит 10^{15}.

Формат выходных данных

Программа должна вывести одно число — ответ, который напечатала бы приведенная выше программа, если бы она была способна работать со столь большими числами.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
25

Стандартный вывод
100

Пример №2

Стандартный ввод
1000000000000000

Стандартный вывод
4500000000000000

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n = int(input())
return str((n//10)*45+(n%10)*(n%10-1)//2)
```

Задача II.1.4.2. Автокорреляционная функция дискретного сигнала (18 баллов)

Темы: математика, реализация.

Условие

Автокорреляционная функция часто применяется при анализе сигналов, например, энцефалограммы человека или в радиолокации. Мы будем рассматривать некоторый цифровой сигнал $a_0, a_1, a_2, \ldots, a_{n-1}$, где каждое значение a_i равно 1 или -1. Определим автокорреляционную функцию $u(t)$ по следующей формуле:

$$u(t) = \sum_{0 \leqslant i < n-t} a_i a_{i+t}.$$

Другими словами, если сигнал задан в виде списка из n значений, то чтобы вычислить автокорреляционную функцию в точке t, требуется взять одну копию списка без первых t элементов, другую копию списка без последних t элементов, поэлементно перемножить эти списки, и найти сумму произведений. Рассмотрим пример. Пусть сигнал содержит шесть элементов $1, 1, -1, 1, -1, 1$. Найдем $u(2)$. Исходная последовательность без первых двух элементов имеет вид $-1, 1, -1, 1$. Исходная последовательность без последних двух элементов имеет вид $1, 1, -1, 1$. Тогда $u(2) = (-1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 2$

По такому же принципу можно посчитать и остальные значения для t от нуля до пяти.

$u(0) = (1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 6$

$u(1) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) + (-1 \cdot 1) + (1 \cdot -1) = -3$

$u(3) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) = -1$

$u(4) = (-1 \cdot 1) + (1 \cdot 1) = 0$

$u(5) = 1 \cdot 1 = 1$

Напишите программу, которая по заданному дискретному сигналу найдет значения автокорреляционной функции для всех t от 0 до $n - 1$.

Формат входных данных

На вход в первой строке подается одно натуральное число n — длина сигнала, $1 \leqslant n \leqslant 100$. Во второй строке через пробел записаны числа $a_0, a_1, \ldots, a_{n-1}$, задающие дискретный сигнал. Каждое значение a_i равно 1 или -1.

Формат выходных данных

Программа должна вывести через пробел n целых чисел — значения автокорреляционной функции $u(0), u(1), \ldots u(n-1)$.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a,end=' ')`.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
6 1 1 -1 1 -1 1

Стандартный вывод
6 -3 2 -1 0 1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
n=int(input())
a=list(map(int,input().split()))
for k in range(n):
    print(sum([x*y for x,y in zip(a[k:],a[:n-k])]),end=' ')
```

Задача II.1.4.3. Три фишки (25 баллов)

Темы: игры.

Условие

Алиса и Боб играют в следующую игру. Имеется игровое поле в виде последовательности клеток, расположенных друг за другом. На поле расположены три фишки, каждая фишка в своей клетке. За один ход каждый игрок должен переместить одну фишку вправо на произвольное ненулевое число клеток. При этом фишка, которой делается ход, не может встать в клетку, где расположена другая фишка или перепрыгнуть через нее. Выигрывает тот игрок, который смог сделать последний ход.

Рассмотрим пример.

Здесь возможны следующие ходы: сместить правую фишку на одну клетку; сместить среднюю фишку на одну клетку; сместить левую фишку на одну, две, три или четыре клетки.

Алиса всегда делает первый ход, а фишки расставляет Боб. Но Боб не хочет побеждать, он хочет, чтобы Алиса нашла выигрышную стратегию. Поэтому он расставляет фишки так, чтобы Алиса могла гарантированно выиграть.

Например, в приведенной выше позиции Алиса должна сместить самую левую фишку на три клетки.

Далее игра зависит от хода Боба. Предположим, он сместит правую фишку на одну клетку. Тогда Алиса в свой ход сместит левую фишку на одну клетку.

Теперь Боб может ходить только средней фишкой. Если он сдвинет ее на одну клетку, то Алиса сдвинет левую фишку на одну клетку.

Бобу остается вновь ходить средней фишкой. Он сдвинет ее на одну клетку, Алиса сдвинет левую фишку на одну клетку и победит.

Для всех других ходов Боба у Алисы также всегда найдется ход, ведущий к победе.

Вы должны написать программу, которая по заданной позиции найдет ход, после которого Алиса сможет победить независимо от дальнейшей игры Боба. Если выигрышных ходов будет несколько, то Алиса может сделать любой из них. Напомним, что исходная позиция будет такой, что найдется как минимум один ход, гарантированно ведущий к победе.

Формат входных данных

На вход подается строка представляющая игровое поле. Пустая клетка в строке обозначена нулем, клетка с фишкой обозначена единицей. Длина строки не превосходит 1000 символов. В строке ровно три единицы.

Формат выходных данных

Программа должна вывести строку, представляющую игровое поле после хода Алисы, в том же формате, в котором она поступает на вход.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
0100001010

Стандартный вывод
0000101010

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
x=list(reversed(input()))
m=[0,0,0]
k=0
for c in x:
    if c=='0':
        m[k]+=1
    elif k<2:
        k+=1
    else:
        break
if m[0]>m[2]:
    x[m[0]]='0'
    x[m[2]]='1'
else:
    x[m[0]+m[1]+m[2]+2]='0'
    x[2*m[0]+m[1]+2]='1'
print(''.join(reversed(x)))
```

Задача II.1.4.4. Выбор купюр (25 баллов)

Темы: реализация.

Условие

В денежной системе Бурляндии выпускаются банкноты всех номиналов от a до $2a$ включительно. У Алисы в бумажнике есть ровно одна банкнота каждого номинала. Алиса хочет сделать покупку ценой b и расплатиться без сдачи. Кроме того, Алиса хочет, чтобы количество потраченных банкнот было как можно меньшим. Напишите программу, которая поможет Алисе выбрать банкноты так, чтобы сумма их номиналов была равна b, а их количество было наименьшим среди возможных. Если указанным условиям удовлетворяет несколько наборов банкнот, то ваша программа может вывести любой из них.

Формат входных данных

На вход в одной строке подается два натуральных числа a и b — минимальный из номиналов купюр и требуемая сумма, $1 \leqslant a \leqslant 100000$. Гарантируется, что для заданной суммы b существует способ получить ее из имеющихся купюр.

Формат выходных данных

Программа должна вывести в одной строке через пробел номиналы всех банкнот, которые потребуются для оплаты. ***Все номиналы должны быть упорядочены по возрастанию.***

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
`10 99`

Стандартный вывод
`10 15 17 18 19 20`

Пояснения к примеру

Сумма чисел, указанных в ответе, равна 99, и все числа лежат в диапазоне от 10 до 20 включительно. При этом сумма номиналов пяти самых ценных банкнот меньше чем 99, поэтому оплатить указанную сумму пятью или меньшим числом банкнот невозможно. Однако другие варианты получения требуемой суммы шестью банкнотами возможны, например, 13 14 15 18 19 20. Такой ответ тоже будет засчитан.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```python
a,b=map(int,input().split())
up,dn,k=0,0,0
while up<b:
    up+=2*a-k
    dn+=a+k
    k+=1

if a>k-1:
    t=(b-dn)//(a-k+1)
    r=(b-dn)%(a-k+1)
else:
    t,r=0,0
if t==k:
    print(*range(2*a-t+1,2*a+1))
else:
    print(*range(a,a+k-t-1),a+k-1-t+r,*range(2*a-t+1,2*a+1))
```

Предметный тур. Математика

Первая попытка. Задачи 8–9 класса

Задача II.2.1.1. (25 баллов)

Темы: вероятность, несовместные события.

Условие

Участникам «математического марафона» предлагается для решения n задач, из которых k имеют повышенную сложность. Задачи решаются по одной в случайном порядке. Участник получает следующую задачу лишь после того, как решил предыдущую. Решив задачу повышенной сложности, участник переходит на следующий уровень. Найти вероятность того, что для выхода на следующий уровень участнику придется решить не более трех задач. Ответ округлить до сотых.

Решение

Событие A, при котором участник решил не более трёх задач до выхода на следующий уровень, можно разбить на три несовместных события:

- A_1 — участник решил ровно одну задачу;
- A_2 — участник решил ровно две задачи;
- A_3 — участник решил ровно три задачи.

В первом случае участник решил одну из k сложных задач, а всего он мог выбрать одну из n задач. Поэтому:

$$P(A_1) = \frac{k}{n}.$$

Во втором случае участник сначала решил простую, а затем сложную задачу. Всего способов выбрать пару задач: $n(n-1)$. Всего способов выбрать сначала простую, а затем сложную задачу: $(n-k)k$. Поэтому:

$$P(A_2) = \frac{(n-k)k}{n(n-1)}.$$

В третьем случае участник сначала решил две простых, а затем сложную задачу. Всего способов выбрать тройку задач: $n(n-1)(n-2)$. Всего способов выбрать сначала две простых, а затем сложную задачу: $(n-k)(n-k-1)k$. Поэтому:

$$P(A_3) = \frac{(n-k)(n-k-1)k}{n(n-1)(n-2)}.$$

Таким образом, искомая вероятность:

$$P(A) = P(A_1) + P(A_2) + P(A_3) = \frac{k}{n} + \frac{(n-k)k}{n(n-1)} + \frac{(n-k)(n-k-1)k}{n(n-1)(n-2)}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	10	20	1
k	3	7	1

Ответ: $\dfrac{k}{n} + \dfrac{(n-k)k}{n(n-1)} + \dfrac{(n-k)(n-k-1)k}{n(n-1)(n-2)}$.

Задача II.2.1.2. (20 баллов)

Темы: *комбинаторика.*

Условие

В вазе лежат конфеты, среди которых нет одинаковых. После обеда мама разрешила своим двум детям взять по три конфеты. Сначала одну конфету берет младший, потом одну конфету берет старший, затем одну конфету опять берет младший и т. д. Известно, что у младшего ребенка число вариантов выбора на n больше, чем у старшего. Сколько конфет лежало в вазе до того, как дети стали их брать?

Решение

Пусть k — количество конфет в вазе.

Младший выбирает сначала одну из k конфет, затем одну из $k-2$, и, наконец, одну из $k-4$. Всего вариантов у младшего ребёнка $k(k-2)(k-4)$.

Старший выбирает сначала одну из $k-1$ конфет, затем одну из $k-3$, и, наконец, одну из $k-5$. Всего вариантов у старшего ребёнка $(k-1)(k-3)(k-5)$.

По условию:

$$k(k-2)(k-4) - (k-1)(k-3)(k-5) = n.$$

Раскрыв скобки, получаем квадратное уравнение:

$$3k^2 - 15k + 15 - n = 0.$$

Решая его и отбрасывая отрицательный корень, находим:

$$k = \frac{15 + \sqrt{45 + 12n}}{6}.$$

Диапазоны

$$n = 3(k^2 - 5k + 5),$$

где $k = 15, ..., 30$.

Ответ: $\dfrac{15 + \sqrt{45 + 12n}}{6}$.

Задача II.2.1.3. (30 баллов)

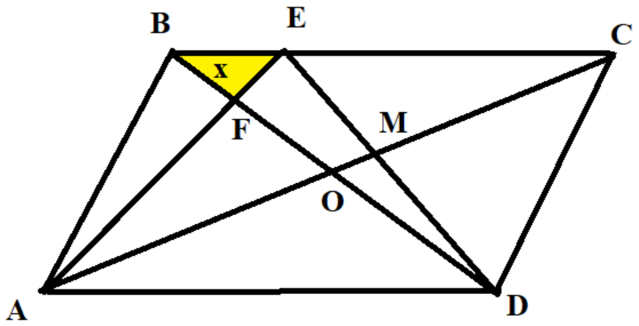
Темы: геометрия, площадь, подобные треугольники.

Условие

В параллелограмме $ABCD$ диагонали пересекаются в точке O. На стороне BC отмечена точка E. Диагональ BD пересекает отрезок AE в точке F так, что площадь треугольника ABF равна a. Диагональ AC пересекает отрезок DE в точке M так, что площадь треугольника DMC равна b. Найдите площадь треугольника BEF, если площадь четырёхугольника $EFOM$ равна c. Ответ округлите до сотых.

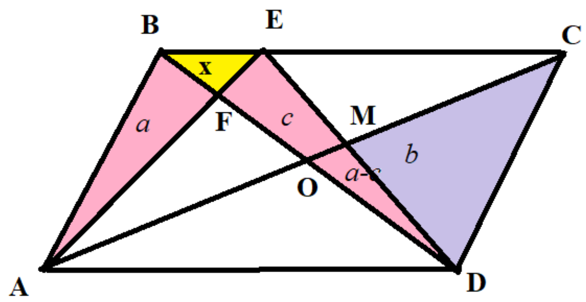
Решение

Обозначим искомую площадь за x.



Рассмотрим треугольники ABE и DBE. Их площади совпадают, следовательно, $a + x = x + c + S_{DOM}$. Отсюда:

$$S_{DOM} = a - c.$$



Поскольку $S_{ABO} = S_{DOC} = b + a - c$, то $S_{AFO} = S_{ABO} - a = b - c$.

Теперь найдём площадь треугольника AFD:

$$S_{AFD} = S_{AFO} + S_{AOD} = S_{AFO} + S_{OCD} = (b - c) + (a - c + b) = 2(b - c) + a.$$

Треугольники BEF и FED имеют одинаковую высоту (из вершины E). Поэтому их площади относятся как основания:

$$\frac{x}{a} = \frac{BF}{FD}.$$

Треугольники ABF и AFD также имеют общую высоту (из вершины A).

Следовательно, отношение их оснований равно отношению их площадей:

$$\frac{BF}{FD} = \frac{a}{2(b-c)+a}.$$

Сопоставляя два последних равенства, находим:

$$x = \frac{a^2}{2(b-c)+a}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	7	15	1
b	8	16	1
c	1	6	1

Ответ: $\dfrac{a^2}{2(b-c)+a}$.

Задача II.2.1.4. (25 баллов)

Темы: *алгебра, система линейных уравнений, задача на работу.*

Условие

Петя и Вася в первый день хакатона совместно за n часов написали программу, решающую предложенную задачу. Через каждые два часа работы над программой мальчики делали короткий перерыв. При этом производительность после перерыва у Пети становилась на 20%, а у Васи — на 10% меньше производительности до перерыва. Во второй день Петя и вовсе потерял интерес к задачам, а Васю, наоборот, заинтересовала предложенная тематика. Поэтому производительность Пети снизилась в k раз, а у Васи она увеличилась в k раз (относительно начала первого дня). Производительность мальчиков падала через каждые два часа так же, как и в первый день. В итоге во второй день они потратили $n-2$ часа на совместное написание кода того же объёма, что и в первый день. Во сколько раз в первый день изначальная производительность Васи отличалась от изначальной производительности Пети? Ответ округлите до сотых.

Решение

Пусть x — производительность Пети, y — производительность Васи. По условию задачи составим систему уравнений:

$$\begin{cases} 2(x+y) + 2(0,8x+0,9y) + (n-4)(0,8^2x+0,9^2y) = 1, \\ 2\left(\frac{x}{k}+ky\right) + (n-4)\left(\frac{0,8x}{k}+0,9ky\right) = 1. \end{cases}$$

Приводя подобные слагаемые, имеем:

$$\begin{cases}(3,6+0,8^2(n-4))x+(3,8+0,9^2(n-4))y=1,\\ \frac{2+0,8(n-4)}{k}x+(2+0,9(n-4))ky=1.\end{cases}$$

Разделив каждое уравнение на x, находим:

$$\begin{cases}3,6+0,8^2(n-4)+(3,8+0,9^2(n-4))\frac{y}{x}=\frac{1}{x},\\ \frac{2+0,8(n-4)}{k}+(2+0,9(n-4))k\frac{y}{x}=\frac{1}{x}.\end{cases}$$

Вычитая уравнения, получаем:

$$(3,8+0,9^2(n-4)-2k-0,9k(n-4))\frac{y}{x}=\frac{2+0,8(n-4)}{k}-3,6-0,8^2(n-4).$$

Отсюда:

$$\frac{y}{x}=\frac{\frac{2+0,8(n-4)}{k}-3,6-0,8^2(n-4)}{3,8+0,9^2(n-4)-2k-0,9k(n-4)}$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	4,5	5	0,5
k	2	5	1

Ответ: $\dfrac{\frac{2+0,8(n-4)}{k}-3,6-0,8^2(n-4)}{3,8+0,9^2(n-4)-2k-0,9k(n-4)}$.

Первая попытка. Задачи 10–11 класса

Задача II.2.2.1. (20 баллов)

Темы: теория вероятностей, формула Байеса.

Условие

Дистрибьютор покупает компьютеры на трех фабриках. Первая фабрика подготовила к отгрузке m_1 компьютеров, k_1 из которых имеют брак. Вторая фабрика — m_2 компьютеров, среди которых k_2 бракованных. Третья фабрика — m_3 компьютеров, среди которых k_3 бракованных. Дистрибьютор приобрел у каждой фабрики по одному компьютеру. Оказалось, что компьютер, купленный покупателем у дистрибьютора, бракованный. Какова вероятность того, что этот компьютер изготовлен j-й фабрикой? Ответ округлить до сотых.

Решение

Пусть событие A — купленный компьютер бракованный, событие H_i — купленный компьютер изготовлен на i-й фабрике. В задаче спрашивается, чему равна условная вероятность $P(H_j|A)$.

Имеем для всех $i = 1, 2, 3$:

$$P(H_i) = \frac{1}{3}, \quad P(A|H_i) = \frac{k_i}{m_i}.$$

Тогда по формуле Байеса

$$P(H_j|A) = \frac{P(H_j)P(A|H_j)}{\sum\limits_{i=1}^{3} P(H_i)P(A|H_i)} = \left(\frac{k_j}{m_j}\right) / \left(\frac{k_1}{m_1} + \frac{k_2}{m_2} + \frac{k_3}{m_3}\right).$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
m_1	6	8	1
k_1	2	3	1
m_2	8	10	1
k_2	2	3	1
m_3	7	9	1
k_3	1	2	1
j	1	3	1

Ответ: $\left(\dfrac{k_j}{m_j}\right) / \left(\dfrac{k_1}{m_1} + \dfrac{k_2}{m_2} + \dfrac{k_3}{m_3}\right)$.

Задача II.2.2.2. (25 баллов)

Темы: комбинаторика, биномиальные коэффициенты.

Условие

На переэкзаменовку пришло a студентов 1-го курса и b студентов 2-го курса. Преподаватель пригласил в аудиторию половину студентов 2-го курса и несколько студентов 1-го курса. Пока студенты готовились, преподаватель подсчитал, что существует c способов вызвать студентов таким образом (то есть половину пришедших второкурсников и количество вызванных им первокурсников). Какое наибольшее возможное число студентов 1-го курса преподаватель мог вызвать, чтобы полученное им число вариантов не изменилось?

Решение

Пусть k — число вызванных преподавателем первокурсников. Тогда число способов вызвать половину второкурсников и k первокурсников:

$$c = C_b^{b/2} C_a^k.$$

Зная число сочетаний $C_a^k = \frac{c}{C_b^{b/2}}$ и число a, найдём $k \leqslant a/2$ (можно, например, воспользоваться треугольником Паскаля).

Так как $C_a^k = C_a^{a-k}$ и $C_a^k \neq C_a^m$ при $m \neq k$ и $m \neq a-k$, то полученное преподавателем число вариантов не изменится, если вызывать $a-k$ первокурсников.

Диапазоны

Величина	min	max	Шаг
a	9	12	1
b	6	12	2
k	3	$\lceil a/2 \rceil - 1$	1
c	$C_b^{b/2} C_a^k$		

Ответ: $a - k$.

Задача II.2.2.3. (30 баллов)

Темы: стереометрия, исследование на экстремум.

Условие

Шар радиуса r см составлен из двух полушаров: из меди плотности 8,9 г/см^3 и олова плотности 7,3 г/см^3. Из этого шара выпилили деталь в форме прямоугольного параллелепипеда наибольшего объёма, стороны основания которого относятся как k:1. Найти массу опилок (то есть всех частей, оставшихся от шара после выпиливания детали).

Принять $\pi = 3,1416$ и выразить ответ в г (с точностью до 1 г).

Решение

Выпиленная деталь представляет собой прямоугольный параллелепипед, вписанный в шар. Её объём:

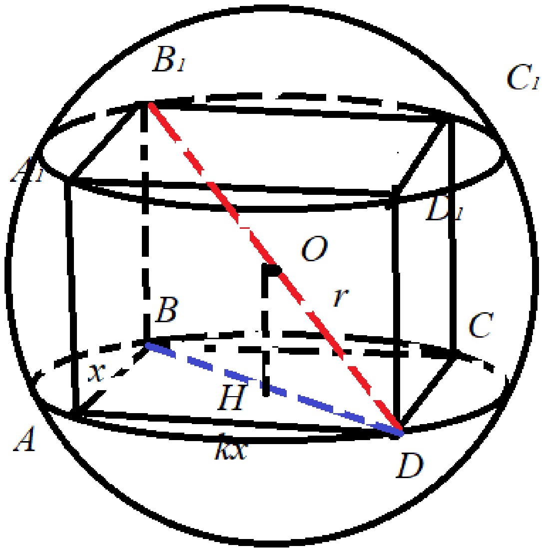
$$V = kx^2 \cdot 2OH.$$

Из треугольников ODH и ABD находим:

$$OH^2 = OD^2 - HD^2 = r^2 - \left(\frac{BD}{2}\right)^2 = r^2 - \frac{x^2 + (kx)^2}{4} = \frac{4r^2 - x^2(k^2+1)}{4}.$$

Тогда:

$$V = kx^2 \cdot 2\sqrt{\frac{4r^2 - x^2(k^2+1)}{4}} = kx^2\sqrt{4r^2 - x^2(k^2+1)}.$$



Найдём максимум функции $V(x)$. Её производная:

$$V'(x) = \frac{2kx(4r^2 - x^2(k^2+1)) - kx^3(k^2+1)}{\sqrt{4r^2 - x^2(k^2+1)}} = kx\frac{8r^2 - 3(k^2+1)x^2}{\sqrt{4r^2 - (k^2+1)x^2}}$$

на промежутке $(0, \frac{2r}{\sqrt{k^2+1}})$ имеет единственный корень:

$$x = \frac{2r\sqrt{2}}{\sqrt{3(k^2+1)}}.$$

Исследование знаков производной показывает, что это точка максимума, а наибольший объём равен:

$$V_{max} = \frac{16kr^3}{3\sqrt{3}(k^2+1)}.$$

Объём опилок равен:

$$V_o = \frac{4\pi r^3}{3} - \frac{16kr^3}{3\sqrt{3}(k^2+1)} = \frac{4r^3}{3}\left(\pi - \frac{4k}{\sqrt{3}(k^2+1)}\right).$$

Поскольку половина этого объёма сделана из олова (плотности $\rho_1 = 7,3$ г/см^3), а половина — из меди (плотности $\rho_2 = 8,9$ г/см^3), то масса опилок равна:

$$M_o = \frac{V_o}{2}\rho_1 + \frac{V_o}{2}\rho_2 = V_o\frac{\rho_1+\rho_2}{2} = 8,1V_0 = 10,8r^3\left(\pi - \frac{4k}{\sqrt{3}(k^2+1)}\right).$$

Погрешность 1.

Диапазоны

Величина	min	max	Шаг
r	2	6	0,2
k	2	10	1

Ответ: $10,8r^3\left(\pi - \frac{4k}{\sqrt{3}(k^2+1)}\right)$.

Задача II.2.2.4. (25 баллов)

Темы: алгебра, функциональное уравнение, линейная система.

Условие

Найти значение $f(1)$ для функции f, удовлетворяющей при всех $x \neq a$ уравнению

$$f(x) + kx \cdot f\left(\frac{ax+b}{x-a}\right) = n.$$

Записать ответ с точностью до 0,01.

Решение

Пусть $y = \frac{ax+b}{x-a}$. Заметим, что $y \neq a$. Поэтому выполнено равенство:

$$f(y) + kyf\left(\frac{ay+b}{y-a}\right) = n.$$

Так как:

$$x = \frac{ay+b}{y-a},$$

получаем:

$$f\left(\frac{ax+b}{x-a}\right) + k\frac{ax+b}{x-a}f(x) = n.$$

Положим $X = f(x)$, $Y = f\left(\frac{ax+b}{x-a}\right)$. Тогда имеем систему уравнений:

$$\begin{cases} X + kxY = n, \\ Y + kyX = n. \end{cases}$$

Решая эту линейную относительно переменных X и Y систему, получаем:

$$X = \frac{n(1-kx)}{1-k^2xy}.$$

Учитывая, что при $x = 1$ будет $y = \dfrac{a+b}{1-a}$, находим $f(1) = \dfrac{n(k-1)(1-a)}{k^2(a+b)-(1-a)}$.

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	20	50	10
b	2	5	1
k	2	5	1
n	100	200	20

Ответ: $\dfrac{n(k-1)(1-a)}{k^2(a+b)-(1-a)}$.

Вторая попытка. Задачи 8–9 класса

Задача II.2.3.1. (20 баллов)

Темы: теория вероятности, дополнительное событие.

Условие

Участникам олимпиады предлагается для решения n задач. Вася может решить каждую из задач с вероятностью p. Найти вероятность того, что Вася решит хотя бы одну задачу. Ответ округлить до сотых.

Решение

Пусть A — событие, при котором Вася решил хотя бы одну задачу. Вычислим вероятность дополнительного события $\overline{A}$ — Вася не решил ни одну задачу:

$$P(\overline{A}) = (1 - p)^n.$$

Тогда:

$$P(A) = 1 - P(\overline{A}) = 1 - (1 - p)^n.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	4	10	1
p	0,1	0,5	0,1

Ответ: $1 - (1 - p)^n$.

Задача II.2.3.2. (25 баллов)

Темы: комбинаторика.

Условие

В спортивном семейном лагере отдыхают n семей, каждая из которых состоит из 5 человек: отца, матери и троих детей. Для предстоящих соревнований с представителями другого спортивного лагеря надо составить команду из пяти человек, в которой должен быть один мужчина, одна женщина, трое детей, но не должно быть родственников. Сколькими способами это можно сделать?

Решение

Сначала выбираем мужчину, для этого имеется n способов. Далее выбираем женщину из оставшихся $n - 1$ семей. Далее выбираем детей из оставшихся $n - 2$ семей:

$3n-6$ способов выбрать первого, $3n-9$ способов — второго (число способов уменьшается на 3, так как запрещено выбирать братьев или сестёр уже выбранного ребёнка), $3n-12$ способов — третьего.

Получается $N = n(n-1)(3n-6)(3n-9)(3n-12)$ вариантов. Однако перестановки выбранных детей не дают новой команды, поэтому полученное число нужно разделить на 3!. Итого имеется:

$$\frac{N}{3!} = \frac{4.5 \cdot n!}{(n-5)!}$$

вариантов.

Диапазоны

Величина	min	max	Шаг
n	5	25	1

Ответ: $\dfrac{4.5 \cdot n!}{(n-5)!}$.

Задача II.2.3.3. (25 баллов)

Темы: планиметрия, прямоугольный треугольник, вписанная окружность.

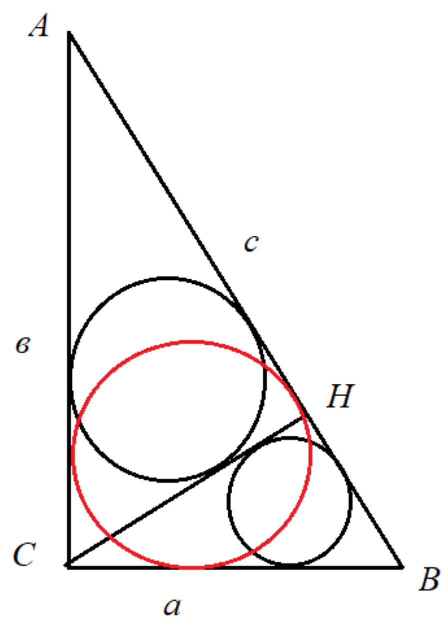
Условие

В прямоугольном треугольнике ABC из вершины прямого угла проведена высота CH. В треугольник ABC вписана окружность радиуса R, а в треугольник ACH — окружность радиуса r. Найти радиус окружности, вписанной в треугольник BCH.

Ввести ответ с точностью до $0,01$.

Решение

Положим $a = BC$, $b = AC$, $c = AB$, x — искомый радиус.



Из подобия треугольников ACH и ABC имеем $\frac{r}{R} = \frac{b}{c}$, следовательно:

$$b = c\frac{r}{R}.$$

Из подобия треугольников BCH и ABC будет $\frac{x}{R} = \frac{a}{c}$, значит:

$$a = c\frac{x}{R}.$$

По теореме Пифагора будет $a^2 + b^2 = c^2$, то есть:

$$\left(c\frac{x}{R}\right)^2 + \left(c\frac{r}{R}\right)^2 = c^2.$$

Отсюда находим:

$$x = \sqrt{R^2 - r^2}$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
r	1	4	0,5
R	6	12	0,5

Ответ: $\sqrt{R^2 - r^2}$.

Задача II.2.3.4. (30 баллов)

Темы: алгебра, поиск шаблона.

Условие

Развозчик пиццы каждому адресату отдавал половину имеющихся пицц и ещё половину пиццы (каждый адресат получал целое число пицц). Все пиццы были розданы по k адресам. Сколько изначально пицц было у развозчика?

Решение

Пусть у разносчика n пицц.

Первому адресату он отдаст $\frac{n}{2} + \frac{1}{2} = \frac{n+1}{2}$ пицц. У него останется $n - \frac{n+1}{2} = \frac{n-1}{2}$ пицц.

Второму адресату он отдаст $\frac{n-1}{4} + \frac{1}{2} = \frac{n+1}{4}$ пицц. У него останется $\frac{n-1}{2} - \frac{n+1}{4} = \frac{n-3}{4}$ пицц.

Третьему адресату он отдаст $\frac{n-3}{8} + \frac{1}{2} = \frac{n+1}{8}$ пицц. У него останется $\frac{n-3}{4} - \frac{n+1}{8} = \frac{n-7}{8}$ пицц. И так далее.

Имеем: k-му адресату он отдаст $\frac{n+1}{2^k}$ пицц. Так как пиццы у него закончатся, то последний адресат должен получить одну пиццу, то есть: $\frac{n+1}{2^k} = 1$.

Следовательно, $n = 2^k - 1$.

Диапазоны

Величина	min	max	Шаг
k	10	20	1

Ответ: $2^k - 1$.

Вторая попытка. Задачи 10–11 класса

Задача II.2.4.1. (25 баллов)

Темы: теория вероятностей, формула полной вероятности.

Условие

В районной «математической регате» участвует n команд, причём физико-математический лицей выставил две команды. По правилам соревнований команды случайным образом разбиваются на игровые пары. В каждой паре определяется проигравшая команда, которая выбывает из турнира, и победившая, которая выходит в следующий тур. На очередном туре команды снова распределяются на пары случайным образом. В каждой встрече вероятность выигрыша и поражения для любой команды равна 0,5. Какова вероятность того, что не позднее k-го тура команды физико-математического лицея сыграют друг с другом? Ответ округлить до тысячных.

Решение

Пусть A — событие, при котором команды физико-математического лицея (назовём их «команда 1» и «команда 2») сыграют вместе не позднее k-го тура; A_1 — событие, при котором эти команды встретятся в первом туре. По формуле полной вероятности:

$$P(A) = P(A|A_1)P(A_1) + P(A|\overline{A}_1)P(\overline{A}_1) = P(A_1) + P(A|\overline{A_1})(1 - P(A_1))$$

(мы воспользовались тем, что $P(A|A_1) = 1$).

Обозначим через $p_{i,j}$ вероятность события, при котором команды 1 и 2 встретились не позднее j-го тура, если всего участвует 2^i команд. Тогда формулу, полученную выше, можно записать так:

$$p_{m,k} = p_{m,1} + P(A|\overline{A_1})(1 - p_{m,1}).$$

Если заранее известно, что произошло событие $\overline{A}_1$, то события «команда 1 выиграла» и «команда 2 выиграла» независимы. Тогда вероятность выхода обеих команд во второй тур равна $0,5^2 = 0,25$. Следовательно, по формуле полной вероятности имеем:

$$P(A|\overline{A_1}) = p_{m-1,k-1} \cdot 0,25 + 0 \cdot (1 - 0,25) = p_{m-1,k-1} \cdot 0,25.$$

Заметим ещё, что:

$$p_{i,1} = \frac{2^i/2}{C^2_{2^i}} = \frac{2^i}{2^i(2^i - 1)} = \frac{1}{2^i - 1}.$$

Таким образом, получаем рекуррентную формулу:

$$p_{m,k} = \frac{1}{2^m - 1} + 0,25p_{m-1,k-1}\left(1 - \frac{1}{2^m - 1}\right).$$

Применяя последовательно это равенство, находим:

$$p_{m,k} = \frac{1}{2^m - 1} + \frac{2(2^{m-1} - 1)}{2^m - 1}\frac{1}{4}\left(\frac{1}{2^{m-1} - 1} + \frac{2(2^{m-2} - 1)}{2^{m-1} - 1}\frac{1}{4}(\ldots)\right) =$$

$$= \frac{1}{2^m - 1} + \frac{1}{2}\frac{1}{2^m - 1} + \frac{1}{4}\frac{1}{2^m - 1} + \ldots = \frac{1}{2^m - 1}\sum_{i=0}^{k-1}\frac{1}{2^i} = \frac{2}{2^m - 1}\left(1 - \frac{1}{2^k}\right).$$

Погрешность 0,1%.

Диапазоны

Величина	min	max	Шаг
n	2^m		
m	5	10	1
k	3	4	1

Ответ: $\dfrac{2}{2^m - 1}\left(1 - \dfrac{1}{2^k}\right).$

Задача II.2.4.2. (20 баллов)

Темы: *комбинаторика.*

Условие

В доме культуры работают 3 различные спортивные секции. Известно, что во второй секции занимается на 3 человека больше, чем в первой, и на 3 человека меньше, чем в третьей, причем спортсмены в каждой секции имеют примерно одинаковый уровень подготовки. С началом сезонных соревнований дом культуры должен каждую неделю посылать в другой район команду из 6-ти человек, в которую должен войти 1 спортсмен из первой секции, 2 спортсмена из второй и 3 спортсмена из третьей (каждый спортсмен участвует в соревнованиях в течение одного сезона один раз). Сколько всего спортсменов занимается в доме культуры, если известно, что число вариантов отбора команды на четвертой неделе равно k?

Решение

Пусть в первой секции занимается n спортсменов. Тогда во второй будет $n+3$, а в третьей $n+6$ спортсменов. На 4-й неделе команду можно составить из $n-1\cdot 3 = n-3$ участников первой секции, $n+3-2\cdot 3 = n-3$ участников второй секции и $n+6-3\cdot 3 = n-3$ участников третьей секции. Таким образом, число вариантов:

$$C^1_{n-3}\cdot C^2_{n-3}\cdot C^3_{n-3} = \frac{(n-3)^3(n-4)^2(n-5)}{12}.$$

Перебирая натуральные значения $n = 6, 7, \ldots$, найдём такое n, что:

$$k = \frac{(n-3)^3(n-4)^2(n-5)}{12}.$$

(Перебор существенно сократится, если рассмотреть разложение числа $12k$ на простые множители.)

Всего спортсменов, занимающихся в доме культуры, равно $n+(n+3)+(n+6) = 3n+9$.

Диапазоны

Величина	min	max	Шаг
n	8	24	1

Ответ: $3n+9$, если $k = \frac{(n-3)^3(n-4)^2(n-5)}{12}$.

Задача II.2.4.3. (30 баллов)

Темы: *стереометрия.*

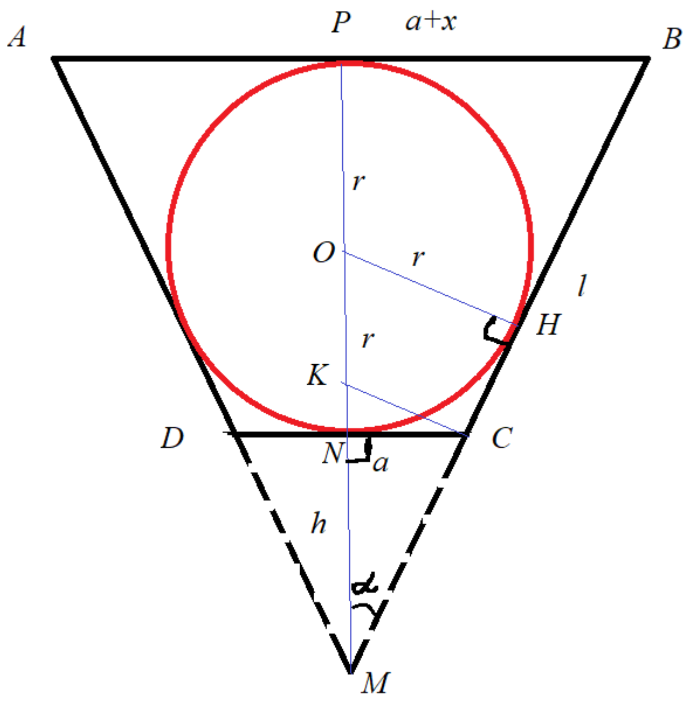
Условие

На дно сосуда с водой, имеющего форму усечённого конуса с радиусом меньшего основания a, положили шар радиуса r (дно соответствует меньшему основанию). При этом оказалось, что шар касается стенок сосуда, а уровень воды касается шара сверху. Какой будет высота столба воды, если вынуть шар?

Ввести ответ с точностью до сотых.

Решение

Пусть радиус поверхности воды равен $a+x$, длина образующей равна l. Достроим имеющийся усечённый конус до полного конуса и рассмотрим сечение полученного тела.



Отрезок AB — уровень воды, равнобедренный треугольник AMB — сечение полного конуса, равнобедренная описанная трапеция $ABCD$ — сечение исходного усечённого конуса. Имеем:

$$\begin{cases} 2l = 2a + 2(a + x), \\ l^2 = x^2 + (2r)^2. \end{cases}$$

Следовательно, $x = \frac{r^2-a^2}{a}$.

Так как треугольники MNC и MOH прямоугольные (O — центр шара), то:

$$\operatorname{tg}\alpha = \frac{a}{h} = \frac{r}{MH}.$$

Имеем:

$$MH^2 = MN \cdot MP = h(h + 2r).$$

Тогда:

$$\frac{a}{h} = \frac{r}{\sqrt{h(h+2r)}}.$$

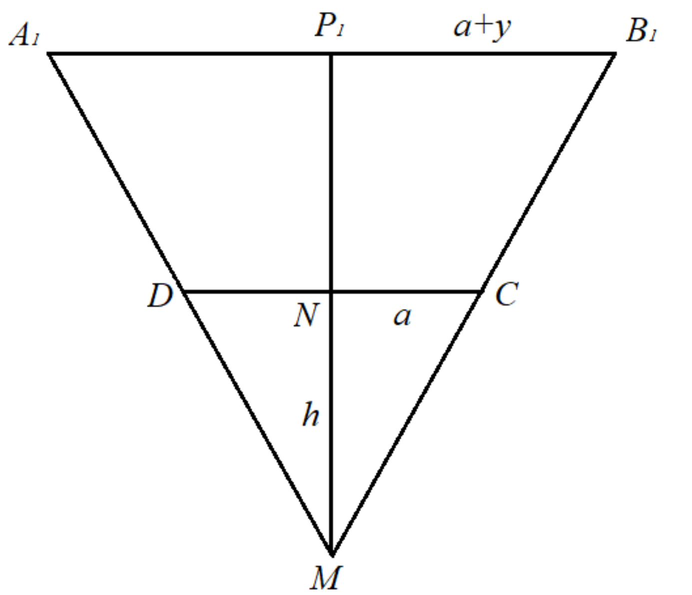
Отсюда:

$$h = \frac{2ra^2}{r^2 - a^2}.$$

Объём конуса равен:

$$V_1 = \frac{1}{3}\pi(a+x)^2(h+2r) = \frac{\pi}{3}\left(a + \frac{r^2 - a^2}{a}\right)^2\left(\frac{2ra^2}{r^2 - a^2} + 2r\right) = \frac{2\pi r^7}{3a^2(r^2 - a^2)}.$$

После вынимания шара имеем следующую картину (A_1B_1 — уровень воды):



Положим $d = MP_1$ (высота нового конуса). Объём нового конуса равен:

$$V_2 = \frac{1}{3}\pi(a+y)^2 d.$$

В то же время:

$$V_2 = V_1 - V_{\text{шара}} = V_1 - \frac{4}{3}\pi r^3.$$

Значит:

$$(a+y)^2 d = \frac{3V_1}{\pi} - 4r^3.$$

Из подобия треугольников MNC и MP_1B_1 находим:

$$\frac{a}{a+y} = \frac{h}{d},$$

поэтому:

$$a + y = \frac{ad}{h}.$$

Следовательно:

$$\left(\frac{ad}{h}\right)^2 d = \frac{3V_1}{\pi} - 4r^3.$$

Подставляя найденные ранее значения h и V_1, получаем:

$$d = \frac{2r}{r^2 - a^2}(r^6 - 2r^2a^2(r^2 - a^2))^{1/3}.$$

Тогда искомая величина NP_1 равна:

$$NP_1 = d - h = \frac{2r}{r^2 - a^2}\left((r^6 - 2r^4a^2 + 2r^2a^4)^{1/3} - a^2\right).$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
r	10	20	0,5
a	4	8	0,5

Ответ: $\dfrac{2r}{r^2 - a^2}\left((r^6 - 2r^4a^2 + 2r^2a^4)^{1/3} - a^2\right)$.

Задача II.2.4.4. (25 баллов)

Темы: алгебра, задача на движение.

Условие

Два друга, Миша и Петя, хотят добраться из пункта A в пункт B. Расстояние между A и B равно S км. В распоряжении ребят есть мопед, передвигающийся со скоростью u км/ч. Отправляясь из A одновременно, Миша идёт пешком, а Петя едет на мопеде до встречи с Васей, идущим из B в A. Дальше Петя идет пешком, а Вася едет на мопеде до встречи с Мишей, после чего передает ему мопед, на котором Миша и добирается до B. За сколько часов до отправления Миши и Пети Вася должен выйти из пункта B, чтобы Миша и Петя добрались до B одновременно? Все три мальчика идут со скоростью v км/ч.

Записать ответ с точностью до 0,01.

Решение

Чтобы Миша и Петя прибыли в пункт B одновременно, они должны пройти пешком одно и то же расстояние x км и проехать на мопеде одно и то же расстояние $S - x$ км. Тогда Вася до встречи с Петей должен пройти x км, и до встречи с Мишей проехать $S - 2x$ км, а затем Миша проедет $S - x$ км. Всего Вася и Миша проедут $2S - 3x$ км. За это время Петя пройдет x км.

Тогда:

$$\frac{2S - 3x}{u} = \frac{x}{v},$$

следовательно:

$$x = \frac{2Sv}{u + 3v}.$$

Петя до встречи с Васей едет $t_1 = \frac{S-x}{u}$ часов. Вася до встречи с Петей идёт $t_2 = \frac{x}{v}$ часов. Тогда Вася должен выйти за $t_2 - t_1$ часов до отправления Миши и Пети:

$$t_2 - t_1 = \frac{x}{v} - \frac{S - x}{u} = \frac{s(u - v)}{u(u + 3v)}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
S	15	30	5
v	5	7	1
u	14	35	1

Ответ: $\dfrac{s(u-v)}{u(u+3v)}$.

Третья попытка. Задачи 8–9 класса

Задача II.2.5.1. (20 баллов)

Темы: вероятность.

Условие

Участникам «математического марафона» предлагается для решения n задач, k из которых — задачи повышенной сложности. Задачи решаются по одной в случайном порядке. Участник получает следующую задачу лишь после того, как решил предыдущую. Решив задачу повышенной сложности, участник переходит на следующий уровень. Найти вероятность того, что для выхода на следующий уровень участнику придётся решить не менее пяти задач. Ответ округлить до сотых.

Решение

Если до выхода на следующий уровень участник решил не менее пяти задач, то первые четыре задачи не имели повышенной сложности, а сложность последующих задач не имеет значения. С учётом порядка всего имеется $n(n-1)(n-2)(n-3)$ способов выбрать первые четыре задачи, и $(n-k)(n-1-k)(n-2-k)(n-3-k)$ способов сделать первые четыре задачи простыми. Поэтому искомая вероятность равна:

$$p = \frac{(n-k)(n-1-k)(n-2-k)(n-3-k)}{n(n-1)(n-2)(n-3)}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	10	15	1
k	3	6	1

Ответ: $\dfrac{(n-k)(n-1-k)(n-2-k)(n-3-k)}{n(n-1)(n-2)(n-3)}$.

Задача II.2.5.2. (25 баллов)

Темы: комбинаторика.

Условие

В танцевальном кружке занимаются m девочек и n мальчиков, которые показывают примерно одинаковые результаты. В отборочных соревнованиях от кружка должны выступать k пар (каждой паре присваивается номер). Сколько существует вариантов составить пары?

Решение

Всего существует C_m^k способов выбрать группу девочек и C_n^k способов выбрать группу мальчиков, которые будут участвовать в соревнованиях. Далее каждому мальчику определим в пару девочку, это можно сделать $k!$ способами. Наконец, нужно присвоить номера парам. Для этого есть $k!$ способов. Итого имеем:

$$C_m^k \cdot C_n^k \cdot k! \cdot k! = \frac{m!n!}{(m-k)!(n-k)!}$$

способов составить пары.

Диапазоны

Величина	min	max	Шаг
n	7	10	1
m	6	7	1
k	3	5	1

Ответ: $\dfrac{m!n!}{(m-k)!(n-k)!}$.

Задача II.2.5.3. (30 баллов)

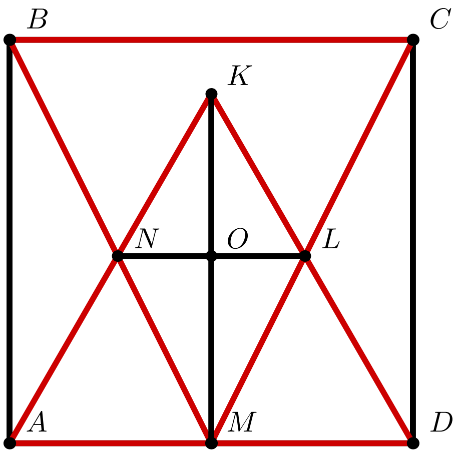
Темы: планиметрия, площадь.

Условие

Точка K лежит внутри квадрата $ABCD$ со стороной a так, что треугольник AKD равносторонний. Точка M делит отрезок AD пополам. Найти площадь фигуры, образованной пересечением треугольников AKD и BCM.

Ввести ответ с точностью до 0,01.

Решение



Пусть $N = BM \cap KA$, $L = CM \cap KD$, $O = NL \cap KM$. Ясно, что $KM \perp NL$. Поэтому площадь фигуры $NKLM$

$$S = S_{NKL} + S_{NML} = \frac{1}{2}KO \cdot NL + \frac{1}{2}OM \cdot NL = \frac{1}{2}KM \cdot NL.$$

Пусть $b = a/2$. По теореме Пифагора из треугольника MKD находим:

$$KM = \sqrt{(2b)^2 - b^2} = b\sqrt{3}.$$

Треугольники ABN и MNK подобны с коэффициентом подобия:

$$k = \frac{2b}{b\sqrt{3}} = \frac{2}{\sqrt{3}}.$$

Следовательно, $AN = k \cdot NK$. Поэтому:

$$2b = AK = AN + NK = k \cdot NK + NK = (k+1)NK.$$

Отсюда

$$NK = \frac{2b\sqrt{3}}{2+\sqrt{3}}.$$

Треугольник NKL равносторонний, значит, $NL = NK$. Имеем:

$$S = \frac{1}{2}KM \cdot NL = \frac{1}{2} \cdot b\sqrt{3} \cdot \frac{2b\sqrt{3}}{2+\sqrt{3}} = \frac{3b^2}{2+\sqrt{3}} = \frac{3a^2}{4(2+\sqrt{3})}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	2	40	4

Ответ: $\dfrac{3a^2}{4(2+\sqrt{3})}$.

Задача II.2.5.4. (25 баллов)

Темы: алгебра, задача на работу.

Условие

Петя, Вася и Иван участвуют в хакатоне по программированию. Если Петя и Вася работают вместе, то их производительности увеличиваются на 10% и 20% соответственно, и они могут написать программу, решающую предложенную задачу, за n часов. Иван же предпочитает индивидуальную работу, поэтому его производительность при совместной работе с Петей падает на 5%, а производительность Пети при таком взаимодействии возрастает на 6%. При совместной работе Ивана и Васи производительность Ивана падает на 15%, а производительность Васи возрастает на 14%. Петя и Иван, работая совместно, могут написать ту же программу за m часов, а Вася и Иван — за k часов. Сколько часов потратят все три мальчика, работая совместно, на написание этой программы, если при таком варианте сотрудничества производительность Пети возрастёт на 8%, Васи — на 17%, а Ивана — уменьшится на 10%? Ответ округлить до сотых.

Решение

Пусть x — производительность Пети, y — производительность Васи, z — производительность Ивана. Принимая за единицу объём работы, необходимый для решения задачи хакатона, получаем систему уравнений:

$$\begin{cases} 1,1x + 1,2y = \frac{1}{n}, \\ 1,06x + 0,95z = \frac{1}{m}, \\ 1,14y + 0,85z = \frac{1}{k}. \end{cases}$$

Складывая все уравнения, имеем:

$$2,16x + 2,34y + 1,8z = \frac{1}{n} + \frac{1}{m} + \frac{1}{k}.$$

Искомое время:

$$t = \frac{1}{1,08x + 1,17y + 0,9z} = \frac{2}{1/n + 1/m + 1/k}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	4	7	1
m	5	8	1
k	6	9	1

Ответ: $\dfrac{2}{1/n + 1/m + 1/k}$.

Третья попытка. Задачи 10–11 класса

Задача II.2.6.1. (25 баллов)

Темы: вероятность, формула Байеса.

Условие

Участникам многопрофильной командной олимпиады предлагаются задачи по трём предметам: математика, информатика и физика. Модуль по математике содержит m задач, k из которых повышенной сложности. Модуль по информатике — n задач, среди которых l повышенной сложности. Модуль по физике — p задач, среди которых r повышенной сложности. Команды получают по три задачи, выбирая из каждого модуля случайно одну задачу. Участники команды π полученные задачи распределили между собой с помощью жеребьёвки, при этом оказалось, что Андрею выпала задача повышенной сложности. Определить вероятность того, что Андрей решает задачу по математике. Ответ округлить до сотых.

Решение

Пусть событие A — Андрею выпала задача повышенной сложности. Определим гипотезы, образующие полную группу событий: H_1 — Андерею выпала задача по математике, H_2 — по информатике, H_3 — по физике. Тогда при $i = 1, 2, 3$:

$$P(H_i) = \frac{1}{3}, \quad P(A|H_1) = \frac{k}{m}, \quad P(A|H_2) = \frac{l}{n}, \quad P(A|H_3) = \frac{r}{p}.$$

По формуле Байеса искомая вероятность равна:

$$P(H_1|A) = \frac{P(H_1)P(A|H_1)}{\sum\limits_{i=1}^{3} P(H_i)P(A|H_i)} = \frac{k/m}{k/m + l/n + r/p}$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
m	8	10	1
k	4	5	1
n	6	8	1
l	3	4	1
p	7	9	1
r	3	4	1

Ответ: $\dfrac{k/m}{k/m + l/n + r/p}$.

Задача II.2.6.2. (20 баллов)

Темы: комбинаторика.

Условие

Сколько различных комбинаций можно составить из n различных цифр, наименьшая из которых в комбинации повторяется ровно k раз, а остальные по одному разу, и первых m букв латинского алфавита, если все цифры должны стоять рядом, а буква «c» должна стоять непосредственно перед буквой «a»?

Решение

Последовательно выбираем:

1. позиции для повторяющейся цифры относительно других цифр — C_{n-1+k}^{k} способов;
2. порядок неповторяющихся цифр — $(n-1)!$ способов;
3. порядок букв — $(m-1)!$ способов (поскольку буква «c» стоит непосредственно перед буквой «a», то можно считать, что у нас имеется $m-1$ буква);
4. положение группы цифр относительно букв — m способов.

Таким образом, всего имеется:

$$C_{n-1+k}^{k}(n-1)!(m-1)!m = \frac{(n-1+k)!m!}{k!}$$

Диапазоны

Величина	min	max	Шаг
n	3	5	1
k	3	5	1
m	5	9	1

Ответ: $\dfrac{(n-1+k)!m!}{k!}$.

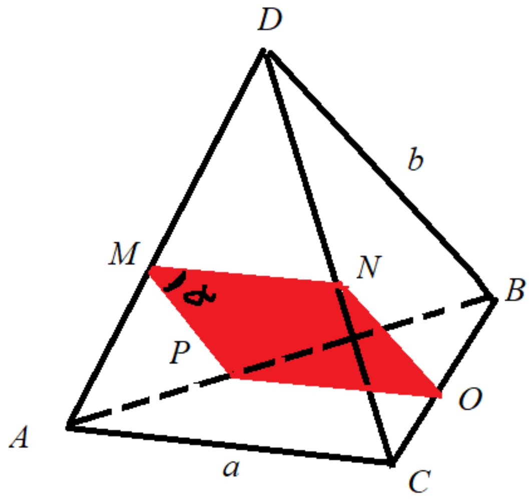
Задача II.2.6.3. (30 баллов)

Темы: стереометрия.

Условие

В тетраэдре $ABCD$ среди сечений, параллельных рёбрам AC и BD одновременно, выбрано сечение с наибольшей площадью. Длина ребра AC равна a, длина ребра BD равна b. Найти сумму квадратов диагоналей этого сечения с точностью до 0,1.

Решение



Пусть $PMNO$ — секущая плоскость. Имеем:

1. $AC, PO \subset ABC$, $AC \parallel MNP \Rightarrow PO \parallel AC$.
2. $BD, NO \subset DBC$, $BD \parallel MNP \Rightarrow NO \parallel BD$.
3. Аналогично, $MN \parallel AC$, $MP \parallel BD$.
4. Из пунктов 1, 2 и 3 следует, что противоположные стороны четырёхугольника $MNOP$ параллельны, а значит, $MNOP$ — параллелограмм.
5. Треугольники PBO и ABC подобны с коэффициентом подобия k. Тогда $PO = MN = ka$.
6. Треугольники APM и ABD подобны с коэффициентом подобия $1-k$. Тогда $MP = NO = (1-k)b$.
7. Угол между прямыми AC и BD равен углу между прямыми MP и MN, обозначим его через α. Площадь сечения равна:
$$S = MP \cdot MN \cdot \sin\alpha = ab\sin\alpha \cdot k(1-k).$$
Отсюда получаем, что площадь сечения максимальна при $k = 1/2$.
8. Стороны сечения равны $PO = MN = \frac{a}{2}$, $MP = NO = \frac{b}{2}$.
9. Сумма квадратов диагоналей параллелограмма равна сумме квадратов всех его сторон:
$$MO^2 + NP^2 = 2(MP^2 + MN^2) = \frac{a^2+b^2}{2}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	4	10	0,5
b	11	20	0,5

Ответ: $\frac{a^2+b^2}{2}$.

Задача II.2.6.4. (25 баллов)

Темы: алгебра, функциональное уравнение.

Условие

Найти значение $f(1)$ для функции f, удовлетворяющей при всех $x \neq a$ уравнению:

$$f(x) + k \cdot f\left(\frac{ax+b}{x-a}\right) = nx.$$

Записать ответ с точностью до 0,01.

Решение

Если $y = \frac{ax+b}{x-a}$, то $x = \frac{ay+b}{y-a}$. Тогда, подставляя вместо x в уравнении значение y, находим:

$$f\left(\frac{ax+b}{x-a}\right) + kf(x) = n \cdot \frac{ax+b}{x-a}.$$

Положим $A = f(x)$, $B = f\left(\frac{ax+b}{x-a}\right)$. Тогда имеем систему уравнений:

$$\begin{cases} A + kB = nx, \\ B + kA = n \cdot \frac{ax+b}{x-a}. \end{cases}$$

Решая её относительно переменных A и B, находим:

$$A = \frac{n(-x^2 + ax(k+1) + kb)}{(x-a)(k^2-1)}.$$

Следовательно:

$$f(1) = \frac{n(a(k+1) + kb - 1)}{(1-a)(k^2-1)}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	10	40	10
b	1	10	1
k	0,1	0,9	0,1
n	5	9	1

Ответ: $\dfrac{n(a(k+1) + kb - 1)}{(1-a)(k^2-1)}$.

Четвертая попытка. Задачи 8–9 класса

Задача II.2.7.1. (25 баллов)

Темы: вероятность.

Условие

Участникам олимпиады предлагается для решения n задач. Вася может решить каждую из задач с вероятностью p. Найти вероятность того, что Вася решит не более одной задачи. Ответ округлить до сотых.

Решение

Вероятность того, что Вася не решит ни одной задачи:

$$p_0 = (1-p)^n.$$

Вероятность того, что Вася решит ровно одну задачу:

$$p_1 = np(1-p)^{n-1}.$$

Значит, вероятность того, что Вася решит не более одной задачи:

$$p_{0,1} = p_0 + p_1 = (1-p)^n + np(1-p)^{n-1}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
n	10	15	1
p	0,1	0,4	0,1

Ответ: $(1-p)^n + np(1-p)^{n-1}$.

Задача II.2.7.2. (20 баллов)

Темы: комбинаторика.

Условие

На спортивные соревнования из другого района приехало n школьников. В гостинице на этот момент оказались свободными только один двухместный номер, один трёхместный номер и один четырёхместный номер. Остальных школьников решили разместить в спортзале местной школы. Руководитель должен составить список на расселение. Сколько возможно вариантов такого списка, если порядок записанных в один номер гостиницы (или в спортзал) не важен?

Решение

Последовательно выбираем спортсменов:

1. в двухместный номер — C_n^2 способов;
2. в трёхместный номер — C_{n-2}^3 способов;
3. в четырёхместный номер — C_{n-5}^4 способов;
4. в спортзал разместятся все остальные спортсмены — 1 способ.

Таким образом, общее число вариантов:

$$C_n^2 C_{n-2}^3 C_{n-5}^4 = \frac{n!}{2!(n-2)!} \frac{(n-2)!}{3!(n-5)!} \frac{(n-5)!}{4!(n-9)!} = \frac{n!}{288 \cdot (n-9)!}.$$

Диапазоны

Величина	min	max	Шаг
n	14	25	1

Ответ: $\dfrac{n!}{288 \cdot (n-9)!}$.

Задача II.2.7.3. (25 баллов)

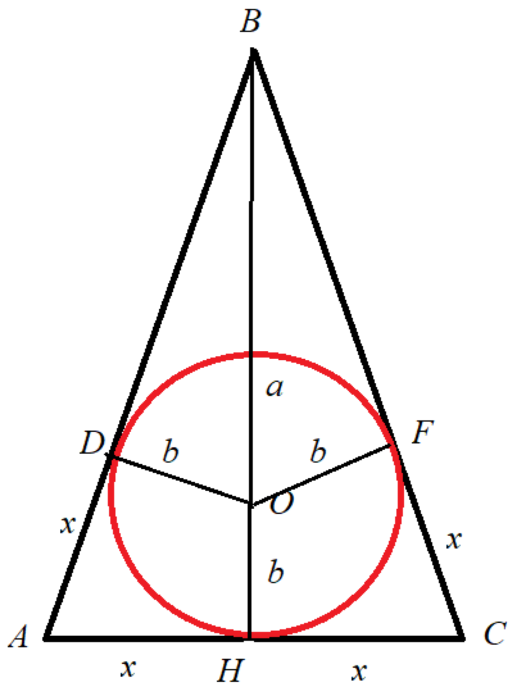
Темы: *планиметрия.*

Условие

В равнобедренном треугольнике ABC с основанием AC вписана окружность с центром в точке O. Точка O делит высоту BH на отрезки длиной a и b, считая от точки B. Найти периметр треугольника ABC.

Ввести ответ с точностью до 0,01.

Решение



Пусть D и F — точки касания окружности с боковыми сторонами треугольника, x — длина AH. Тогда $AD = AH = CH = CF = x$.

Из треугольника OBD по теореме Пифагора получаем $BD^2 = a^2 - b^2$. Следовательно, в треугольнике ABH: $(\sqrt{a^2 - b^2} + x)^2 = (a + b)^2 + x^2$.

Отсюда $x = b\frac{a+b}{\sqrt{a^2-b^2}}$.

Периметр треугольника ABC равен:

$$4x + 2\sqrt{a^2 - b^2} = \frac{2(a + b)^2}{\sqrt{a^2 - b^2}}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	10	20	1
b	2	9	1

Ответ: $\dfrac{2(a + b)^2}{\sqrt{a^2 - b^2}}$.

Задача II.2.7.4. (30 баллов)

Темы: *алгебра, поиск шаблона.*

Условие

Петя приехал на курорт на k дней, имея на банковской карте целое число тысяч рублей. Каждый день он тратил половину имеющихся денег и еще 500 рублей, причём остаток на его счете в конце дня всегда делился на тысячу нацело. В последний день были потрачены все деньги. Сколько изначально рублей было у Пети?

Решение

Пусть у Пети n тыс. руб.

В первый день он потратил $\frac{n}{2}+\frac{1}{2}=\frac{n+1}{2}$ тыс. руб. У него осталось $n-\frac{n+1}{2}=\frac{n-1}{2}$ тыс. руб.

Во второй день он потратил $\frac{n-1}{4}+\frac{1}{2}=\frac{n+1}{4}$ тыс. руб. У него осталось $\frac{n-1}{2}-\frac{n+1}{4}=\frac{n-3}{4}$ тыс. руб.

В третий день он потратил $\frac{n-3}{8}+\frac{1}{2}=\frac{n+1}{8}$ тыс. руб. У него осталось $\frac{n-3}{4}-\frac{n+1}{8}=\frac{n-7}{8}$ тыс. руб.

И так далее: в j-ый день он потратит $\frac{n+1}{2^j}$ тыс. руб.

В последний день он потратит одну тыс. руб., то есть:

$$\frac{n+1}{2^k}=1.$$

Поэтому $n=2^k-1$. Тогда всего рублей $1000(2^k-1)$.

Диапазоны

Величина	min	max	Шаг
k	6	15	1

Ответ: $n=1000(2^k-1)$.

Четвертая попытка. Задачи 10–11 класса

Задача II.2.8.1. (25 баллов)

Темы: теория вероятностей.

Условие

В районной «математической регате» участвует 2^m команд, причём физико-математический лицей выставил две команды. По правилам соревнований в каждом туре команды случайным образом разбиваются на игровые пары. В каждой паре определяется проигравшая команда, которая выбывает из турнира, и победившая, которая выходит в следующий тур. В каждой встрече вероятность выигрыша и поражения для любой команды равна 0,5. Какова вероятность того, что команды физико-математического лицея сыграют друг с другом в k-м туре?

Ответ округлить до тысячных.

Решение

Пусть событие A_k — команды физико-математического лицея встретились на k-м туре; событие B_k — команды физико-математического лицея прошли в k-й тур. Имеем следующую цепочку вложений:

$$\overline{A}_1 \supset B_2\overline{A}_2 \supset B_3\overline{A}_3 \supset \ldots \supset B_{k-1}\overline{A}_{k-1} \supset B_k \supset A_k,$$

поэтому

$$A_k = \overline{A}_1 B_2 \overline{A}_2 B_3 \cdot \ldots \cdot \overline{A}_{k-1} B_k A_k.$$

По формуле для вероятности произведения получаем:

$$P(A_k) = P(\overline{A}_1)P(B_2|\overline{A}_1)P(\overline{A_2}|\overline{A}_1 B_2) \cdot \ldots \cdot P(A_k|\overline{A}_1 B_2 \overline{A}_2 B_3 \cdot \ldots \cdot \overline{A}_{k-1} B_k)$$

Последовательно находим:

$$P(\overline{A}_1) = 1 - \frac{2^m/2}{C_{2^m}^2} = \frac{2(2^{m-1}-1)}{2^m-1},$$

$$P(B_2|\overline{A_1}) = 0,5 \cdot 0,5 = \frac{1}{4},$$

$$P(\overline{A_2}|\overline{A}_1 B_2) = \frac{2(2^{m-2}-1)}{2^{m-1}-1} \quad \text{(вычисляется аналогично } P(\overline{A}_1)),$$

$$P(B_3|\overline{A}_1 B_2 \overline{A}_2) = 0,5 \cdot 0,5 = \frac{1}{4},$$

$$\ldots$$

$$P(\overline{A}_{k-1}|\overline{A}_1 B_2 \overline{A}_2 B_3 \cdot \ldots \cdot B_{k-1}) = \frac{2(2^{m-(k-1)}-1)}{2^{m-(k-1)+1}-1},$$

$$P(B_k|\overline{A}_1 B_2 \overline{A}_2 B_3 \cdot \ldots \cdot B_{k-1}\overline{A}_{k-1}) = 0,5 \cdot 0,5 = \frac{1}{4},$$

$$P(A_k|\overline{A}_1 B_2 \overline{A}_2 B_3 \cdot \ldots \cdot \overline{A}_{k-1} B_k) = \frac{2^{m+1-k}/2}{C_{2^{m+1-k}}^2} = \frac{1}{2^{m+1-k}-1}$$

Перемножая полученные значения, получаем:

$$P(A_k) = \frac{2(2^{m-1}-1)}{2^m-1} \cdot \frac{1}{4} \cdot \frac{2(2^{m-2}-1)}{2^{m-1}-1} \cdot \frac{1}{4} \cdot \ldots \cdot \frac{2(2^{m-(k-1)}-1)}{2^{m-(k-1)+1}-1} \cdot \frac{1}{4} \cdot \frac{1}{2^{m+1-k}-1}$$
$$= \frac{1}{2^{k-1}(2^m-1)}.$$

Погрешность 0,1%.

Диапазоны

Величина	min	max	Шаг
m	5	7	1
k	3	5	1

Ответ: $\dfrac{1}{2^{k-1}(2^m-1)}$.

Задача II.2.8.2. (20 баллов)

Темы: комбинаторика, сочетания с повторениями.

Условие

Команде из k горнолыжников перед соревнованием на новой трассе предоставляется возможность совершить n тренировочных спусков. Тренер должен указать, сколько тренировочных спусков предоставляется каждому спортсмену его команды. Сколько существует вариантов такого распределения, если известно, что каждый спортсмен должен совершить не менее m тренировочных спусков?

Решение

Предоставив m тренировочных спусков каждому спортсмену, останется распределить $n-km$ спусков. Это схема выбора $j = n-km$ предметов из $l = k$ без упорядочения и с повторениями. Количество всевозможных таких наборов — число сочетаний с повторениями:

$$\overline{C}_l^j = C_{l+j-1}^j.$$

Подставляя значения l и j, находим:

$$\overline{C}_k^{n-km} = C_{k+n-km-1}^{n-km} = \frac{(k+n-km-1)!}{(n-km)!(k-1)!}.$$

Диапазоны

Величина	min	max	Шаг
n	60	100	1
k	4	7	1
m	5	7	1

Ответ: $\dfrac{(k+n-km-1)!}{(n-km)!(k-1)!}$.

Задача II.2.8.3. (30 баллов)

Темы: стереометрия.

Условие

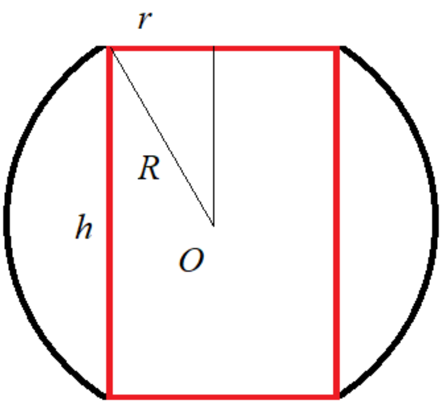
В деревянном шаре радиуса R плотности ρ_1 просверлили сквозное цилиндрическое отверстие (ось цилиндра проходит через центр шара) и заполнили образовавшуюся полость металлическим сплавом плотности ρ_2. Каков наибольший радиус этого отверстия, при котором полученное тело не утонет в жидкости плотности ρ_3?

Примечание: плотность тела — это отношение массы тела к его объёму. Из закона Архимеда следует, что тело не тонет, если его плотность не превосходит плотности жидкости.

Записать ответ с точностью до 0,01.

Решение

Пусть r — искомый радиус цилиндра, h — высота цилиндра. Изобразим сечение полученного тела плоскостью, проходящей через ось цилиндра.



Плотность полученного тела, имеющего массу M и объём V, равна:

$$\rho = \frac{M}{V} = \frac{V_1\rho_1 + V_2\rho_2}{V_1 + V_2},$$

где V_1 — объём деревянной части, V_2 — металлической.

Объём цилиндра равен:

$$V_2 = h \cdot \pi r^2.$$

Для нахождения объёма деревянной части вычтем из шарового слоя объём цилиндра:

$$V_1 = V_{\text{шар. сл.}} - V_2 = \pi h(r^2 + h^2/6) - h\pi r^2 = \frac{\pi h^3}{6}.$$

Условие, при котором тело не тонет, имеет вид $\rho \leqslant \rho_3$, то есть:

$$\frac{\frac{\pi h^3}{6}\rho_1 + h\pi r^2\rho_2}{\pi h(r^2 + h^2/6)} \leqslant \rho_3,$$

что равносильно:

$$r^2(\rho_2 - \rho_3) \leqslant \frac{h^2}{6}(\rho_3 - \rho_1).$$

Подставляя $h^2 = 4(R^2 - r^2)$ (из теоремы Пифагора), после несложных преобразований приходим к неравенству:

$$r \leqslant R\sqrt{\frac{2(\rho_3 - \rho_1)}{3\rho_2 - \rho_3 - 2\rho_1}}.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
R	2	10	1
ρ_1	0,5	0,9	0,1
ρ_2	7	12	1
ρ_3	1	4	0,5

Ответ: $R\sqrt{\dfrac{2(\rho_3-\rho_1)}{3\rho_2-\rho_3-2\rho_1}}$.

Задача II.2.8.4. (25 баллов)

Темы: алгебра, функциональное уравнение.

Условие

Дима и Лёша хотят попасть из пункта A в пункт B. Расстояние между A и B равно S км. В распоряжении ребят есть скейтборд, скорость передвижения на котором равна u км/ч. Отправляясь одновременно, Дима идёт пешком, а Лёша едет на скейтборде до встречи с Пашей, который вышел из B в A за t часов до отправления Димы и Лёши. Дальше Лёша идёт пешком, а Паша едет на скейтборде до встречи с Димой, после чего передаёт ему скейтборд, на котором тот и добирается до B. Известно, что Дима и Лёша добрались до B одновременно. Чему равна скорость ходьбы мальчиков (в км/ч), если известно, что она у всех одинаковая?

Записать ответ с точностью до 0,01.

Решение

Пусть скорость мальчиков равна v км/ч. Чтобы Дима и Лёша прибыли в пункт B одновременно, они должны пройти пешком одно и то же расстояние x км и проехать на скейтборде одно и то же расстояние $S-x$ км. Тогда Паша до встречи с Лёшей должен пройти x км, и до встречи с Димой проехать $S-2x$ км, а затем Дима проедет $S-x$ км. Всего Паша и Дима проедут $2S-3x$ км. За это время Лёша пройдёт x км.

Тогда:

$$\frac{2S-3x}{u}=\frac{x}{v}.$$

Лёша до встречи с Пашей едет $\frac{S-x}{u}$ часов. Паша до встречи с Лёшей идёт $\frac{x}{v}$ часов.

Тогда:

$$\frac{x}{v}-\frac{S-x}{u}=t.$$

Из системы:

$$\begin{cases}\frac{2S-3x}{u}=\frac{x}{v},\\ \frac{x}{v}-\frac{S-x}{u}=t\end{cases}$$

находим $v=\dfrac{u(S-ut)}{S+3ut}$.

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
S	30	40	5
u	10	12	1
t	0,25	0,75	0,25

Ответ: $\dfrac{u(S-ut)}{S+3ut}$.

Инженерный тур

Задача II.3.1. (100 баллов)

Темы: искусственный интеллект, машинное обучение.

Особенности начисления баллов

Так как задача по машинному обучению не предполагает конкретного итогового алгоритма/ответа, а качество решения определяется по значению определенной метрики относительно решений других участников в рейтинге, конвертация баллов в 100-балльную систему может осуществляться только после завершения соревнования с учетом итоговых результатов, полученных участниками и их места в рейтинге.

Условие

Исследование видимого спектра космических объектов — и в первую очередь звёзд — основной способ получения учеными сведений о том, что происходит в дальнем космосе.

Однако каждая космическая фотография содержит огромное количество объектов. Анализ снимков вручную перегружает исследовательские группы. Это скучная и рутинная работа, отвлекающая учёных от более масштабных и творческих исследований. Искусственный интеллект может им в этом помочь.

Ваша задача — создать алгоритм, который на основании предоставленных данных определяет, является наблюдаемая звезда гигантом или карликом.

Описание данных

Данные соревнования сгенерированы на основе открытых данных:

- `Vmag` — видимая звёздная величина;
- `Plx` — расстояние между звездой и Землей;
- `e_Plx` — стандартная ошибка для `Plx`;
- `B-V` — показатель цвета звезды;
- `SpType` — спектральный класс звезды согласно классификации Моргана – Кинана;
- `Amag` — абсолютная магнитуда звезды;
- `TargetClass` — является ли звезда карликом (0) или гигантом (1).

Метрика

Это задача бинарной классификации, и в качестве метрики была выбрана ROC-AUC.

В этом ноутбуке (https://colab.research.google.com/drive/11GhF-zjjG0a67St703KjL9p_iSL8zhRb?usp=sharing) представлен фрагмент кода для расчета метрики и оценки алгоритмов участников.

```
""" Здесь написан фрагмент кода, который позволит рассчитать метрику и
↪  визуализировать её. Примените данный фрагмент в своей работе, предварительно
↪  адаптировав его.

Здесь testy - значения целевой переменной на тестовой выборке, a probs -
↪  предсказанные алгоритмом значения целевой переменной.
"""
import numpy as np
import pandas as pd
from sklearn.metrics import roc_auc_score, roc_curve

# рассчитываем ROC AUC
auc = roc_auc_score(testy, probs)
print('ROC AUC=%.3f' % (auc))

# рассчитываем roc-кривую
fpr, tpr, treshold = roc_curve(testy, probs)
roc_auc = auc(fpr, tpr)

# строим график
plt.plot(fpr, tpr, color='darkorange',
         label='ROC кривая (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Пример ROC-кривой')
plt.legend(loc="lower right")
plt.show()
```

Решение

Решением задачи будет файл с предсказанием, загружаемый на платформу проведения соревнования для оценки качества решения. Пример загружаемого файла с решением (см. таблицу).

Vmag	Plx	e_Plx	B-V	SpType	Amag	TargetClass
7.968	3.094862639	1.012046106	0.9994262014	G8/K0III+A/F	15.20585449	1
7.655	4.749851734	0.9769510133	1.019294078	K0III	16.3322263	0
8.08707314	2.723313345	1.082292443	1.200966303	K1III	14.97541512	1
2.823812958	16.14067008	0.7543589691	0.5387466255	A7III	16.35256883	0
8.499184279	3.936000557	1.747526107	0.9640262152	K0III	16.3815421	1
7.79944689	6.689898891	1.128250145	0.9354923859	K2III	17.8664467	0

Работа наставника НТО на втором отборочном этапе

На втором отборочном этапе участникам предлагаются индивидуальные и командные задачи в рамках выбранных профилей. Для подготовки к нему наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Подготовка по образовательным программам НТО по ряду технологических направлений.
- Разбор задач второго отборочного этапа НТО прошлых лет.
- Прохождение онлайн-курсов по разбору задач НТО прошлых лет.
- Прохождение онлайн-курсов, рекомендованных разработчиками профилей.
- Разбор материалов для подготовки к профилям.
- Практикумы. Для организации практикумов возможно использовать разные подходы или их комбинации:
  - Проведение практикумов по описаниям на страницах профилей и материалов для подготовки.
  - Декомпозиция задач заключительных этапов прошлых лет для выделения наиболее актуальных элементов и их изучения.
  - Анализ технических знаний и навыков (hard skills), требуемых для конкретного профиля, и самостоятельная разработка или поиск занятия для развития наиболее актуальных из них.
  - Посещение практикумов на площадках подготовки и онлайн-мероприятий от разработчиков профилей. Объявления о таких мероприятиях публикуются в группах НТО в VK и в телеграм-канале для наставников НТО (https://t.me/kruzhok_association).

Второй отборочный этап

Задача IV.1. Классификация видео (100 баллов)

Данная задача позволяла участникам освоить основы работы с видео как источником данных для анализа и адаптироваться к особенностям формата.

Задачи классификации, к которым относится и эта, являются важнейшим классом задач обучения с учителем (`supervised learning`). Задача классификации коротких видеофрагментов имеет широкое практическое применение, а успешные модели могут быть использованы, например, для фильтрации видеоконтента или тематической классификации видео, а также в задаче поиска (`information retrieval`).

Условие

В рамках задачи 2-го этапа участникам нужно было разработать алгоритм, который распределит короткие видео (< 1 минуты) на 9 заданных классов:

1. `animal` — в видео присутствует одно или несколько животных;
2. `car` — в видео присутствуют автомобили;
3. `cloud` — в видео показано небо в облаках (тучах);
4. `dance` — в видео есть танцующий человек (люди);
5. `fire` — в видео присутствует пламя (огонь, пламя свечи и т. д.);
6. `flower` — в видео присутствуют цветы;
7. `food` — в видео готовят еду, либо просто присутствует еда;
8. `sunset` — в видео показан закат или рассвет;
9. `water` — в видео показана поверхность воды.

Для работы участникам был предоставлен обучающий набор данных, который включает в себя 864 коротких видео и метки, соответствующие этим видео.

Критерии оценивания

Обученные модели участников запускались на тестовых данных, формат которых соответствует обучающей выборке, однако сами данные были скрыты от участников.

Тестовые данные представлены двумя наборами: `Public` и `Private test`, на которых была рассчитана метрика `accuracy`.

Метрика `accuracy` является стандартной метрикой для задач классификации. Она характеризует долю точных совпадений среди пар предсказанных и истинных ответов, то есть отражает отношение числа совпавших ответов (когда модель участника предсказала такой же ответ, как истинный) к общему числу ответов (независимо от того, в определении какого именно класса модель допустила ошибку). Эта метрика изменяется от 0 до 1, где 0 — наихудшее значение, 1 — наилучшее.

Метрика, рассчитанная на данных `Public test`, была доступна участникам в процессе соревнования на платформе. Финальное значение метрики было рассчитано на данных `Private test`, на основании которого был сформирован итоговый рейтинг участников.

Результаты итогового рейтинга участников были переведены в стобалльную систему в соответствии с формулой, где участник, занявший первое место рейтинга, получал 100 баллов:

$$Total = \frac{Result \times (RankParticipants + 1 - RankPlace) \times 100}{MaxResult \times Participants},$$

где $Total$ — итоговый балл за решение задачи;

$Result$ — результат решения задачи;

$RankParticipants$ — общее количество участников в рейтинге;

$RankPlace$ — место участника в рейтинге;

$MaxResult$ — максимальный результат решения задачи среди всех участников;

$Participants$ — общее количество участников.

Решение

Базовое решение от разработчиков: https://storage.yandexcloud.net/ds-ods/files/materials/93627f7d/Baseline.ipynb.

Пример загружаемого на платформу файла: https://storage.yandexcloud.net/ds-ods/files/materials/09d4f50a/submit.zip.

В проверяющую систему было необходимо отправить код алгоритма, запакованный в ZIP-архив. Решения запускались в изолированном окружении при помощи Docker. Время и ресурсы во время тестирования были ограничены:

- 5Gb на архив с решением;
- 25 минут на работу решения.

Так как решение не имеет доступ к Интернету, все дополнительные данные, например, веса обученной модели, должны были быть подгружены в контейнер. В качестве примера организаторы предоставляли контейнер для запуска бейзлайн-решения.

Доступные ресурсы для проверки решения:

- 8 ядер CPU;
- 48Gb RAM;
- видеокарта NVidia Tesla V100.

Разбор базового решения от разработчиков: https://vk.com/video-165533067_456239151.

Работа наставника НТО при подготовке к заключительному 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. Для подготовки рекомендуется использовать сборники задач прошлых лет. Кроме того, наставнику важно изучить организационные особенности заключительного этапа, чтобы помочь ученикам разобраться в формальных особенностях его проведения.

Наставник НТО также может познакомиться с разработчиками профилей для получения консультации о подготовке к заключительному этапу, дополнительных материалах и способах поддержки высокой мотивации участников.

При работе с командой участников рекомендуется уделить внимание следующим вопросам:

- Сплочение команды. Наставнику необходимо уделить этому особенное внимание, если участники команды находятся в разных городах и не имеют возможности встретиться в очном формате. Регулярные встречи, в том числе в дистанционном формате, помогут поддержать эффективную и позитивную коммуникацию внутри команды.
- Анализ состава команды. Необходимо обсудить роли участников в команде и задачи, которые им предстоит решать в рамках выбранных ролей. Кроме того, нужно обсудить взаимозаменяемость ролей.
- Анализ знаний и компетенций участников. Необходимо убедиться, что участники обладают нужными навыками и компетенциями и продумать план по формированию и развитию недостающих навыков и компетенций.
- Составление плана подготовки. График занятий строится, исходя из даты начала заключительного этапа.
- Участие в подготовительных мероприятиях от разработчиков профилей. Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, хакатоны, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.
- Проведение практикумов или хакатонов. Для этого наставники могут использовать материалы для подготовки к соответствующему профилю и сборники задач прошлых лет. Практикумы и хакатоны могут проводиться дистанционно, рекомендации для этого формата приведены в сборниках 2020–22 гг.

Во время заключительного этапа участников сопровождают модераторы или вожатые, разработчики профиля и организаторы НТО. Внешнее вмешательство в ход соревнований запрещено. Участники, получившие во время проведения НТО стороннюю помощь, могут быть дисквалифицированы.

Заключительный этап

Предметный тур

Информатика. 8–11 класс

Задача VI.1.1.1. Покупка серверов (10 баллов)

Условие

Для обучения искусственного интеллекта (ИИ) требуются вычислительные мощности. Компания, в которой вы работаете, поручила вам купить K серверов, на которых будут обучаться нейросети.

На выбор есть N различных серверов, про каждый известна его цена C_i и его мощность B_i. Выберите K наилучших серверов по соотношению мощность/цена. Можно купить не более одного сервера каждого вида.

Формат входных данных

В первой строке даны 2 числа — N и K, про которые известно, что $(1 \leqslant N \leqslant 100000)$, $(1 \leqslant K \leqslant N)$.

В следующей строке даны N чисел B_i — мощности серверов, про которые известно, что $(1 \leqslant B_i \leqslant 10000)$.

В третьей строке даны N чисел C_i — цены серверов, про которые известно, что $(1 \leqslant C_i \leqslant 10000)$.

Формат выходных данных

Выведите **по возрастанию** K различных чисел из диапазона $[1...N]$ — номера купленных серверов. Если оптимальных ответов несколько, вывести любой.

Примеры

Пример №1

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

Стандартный вывод
1 2 3 5

Решение

Отсортируем компьютеры по соотношению мощность/цена и выберем k наилучших.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;

bool cmp(vector<int>& l, vector<int>& r) {
    int left = l[1] * r[2];
    int right = l[2] * r[1];
    return left < right;
}

int main() {
    int n, k;
    cin >> n >> k;
    vector<int> powers(n);
    for (int i = 0; i < n; ++i)
        cin >> powers[i];
    vector<int> costs(n);
    for (int i = 0; i < n; ++i)
        cin >> costs[i];

    vector<vector<int>> servers(n);
    for (int i = 0; i < n; ++i) {
        servers[i].resize(3);
        servers[i][0] = i + 1;
        servers[i][1] = powers[i];
        servers[i][2] = costs[i];
    }
    sort(servers.begin(), servers.end(), cmp);
    reverse(servers.begin(), servers.end());

    vector<int> ans(k);
    for (int i = 0; i < k; ++i)
        ans[i] = servers[i][0];
    sort(ans.begin(), ans.end());
    for (int i = 0; i < k; ++i)
        cout << ans[i] << " ";
    cout << '\n';
}
```

Задача VI.1.1.2. Подбор картинки (15 баллов)

Условие

Ученые загадочной страны Берляндии разработали новую модель ИИ, предсказывающую насколько картинка понравится пользователю. Согласно разработанной теории, картинка — это набор пикселей на плоскости, а удовольствие от ее просмотра прямо пропорционально количеству одноцветных тупоугольных треугольников.

Помогите ученым в написании этой модели. Ваша задача состоит в том, чтобы посчитать, сколько одноцветных тупоугольных треугольников есть в изображении.

В данной задаче мы считаем, что пиксели настолько малы, что они не воспринимаются как точки на плоскости. Считаем координатами точки-пикселя номер строки и столбца, в которых он находится.

Треугольник называется одноцветным, если во всех трех его вершинах стоят одинаковые цвета.

Треугольник называется тупоугольным, если один из его углов строго больше 90°.

Треугольник называется вырожденным, если все три его вершины расположены на одной прямой.

Вырожденные треугольники считать не нужно.

Формат входных данных

В первой строке даны 2 числа — N и M, про которые известно, что $(1 \leqslant N \leqslant 10)$, $(1 \leqslant M \leqslant 10)$.

В следующих N строках дано по M чисел через пробел — цвета соответствующих пикселей на картинке.

Каждый пиксель — это число X, про которое известно, что $(0 \leqslant X \leqslant 256^3)$.

Формат выходных данных

Выведите одно число — количество тупоугольных треугольников на картинке. Если их нет — вывести 0.

Примеры

Пример №1

Стандартный ввод
2 3 2 2 1 2 3 2

Стандартный вывод
1

Пояснения к примеру

На картинке всего один пиксель цвета 1 и один пиксель цвета 3, треугольников они не образуют.

Пиксели цвета 2 имеют координаты:

(0, 0) — левый верхний (1).

(1, 0) — левый нижний (2).

(0, 1) — средний столбцей верхний (3).

(1, 2) — правый нижний (4).

Направление осей: ось X направлена вниз, ось Y направлена вправо. Если записать их в двумерный массив, то пиксель с координатами (x, y) можно получить как `matrix[x][y]`. Всего есть 4 треугольника — (1, 2, 3), (1, 2, 4), (2, 3, 4) и (1, 3, 4). Первые 3 являются прямоугольными, последний — тупоугольный.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;

int len(int x1, int y1, int x2, int y2) {
    int dx = x1 - x2;
    int dy = y1 - y2;
    return dx * dx + dy * dy;
}

bool check_obtuce_triangle(int x1, int y1, int x2, int y2, int x3, int y3) {
    int a = len(x1, y1, x2, y2);
    int b = len(x1, y1, x3, y3);
    int c = len(x2, y2, x3, y3);
    vector<int> tmp = {a, b, c};
    sort(tmp.begin(), tmp.end());
    a = tmp[0]; b = tmp[1]; c = tmp[2];
    if ((x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1) == 0)
        return false; // вырожденный треугольник
    if (a + b < c)
        return true;
    return false;
}

int main() {
    int n, m;
    cin >> n >> m;
    vector<vector<int>> data(n, vector<int>(m));
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j)
            cin >> data[i][j];
    int ans = 0;
    for (int x1 = 0; x1 < n; ++x1)
        for (int y1 = 0; y1 < m; ++y1)
            for (int x2 = 0; x2 < n; ++x2)
                for (int y2 = 0; y2 < m; ++y2)
                    for (int x3 = 0; x3 < n; ++x3)
                        for (int y3 = 0; y3 < m; ++y3)
                            if (data[x1][y1] == data[x2][y2] && data[x1][y1] ==
                            ↪ data[x3][y3])
                                if (check_obtuce_triangle(x1, y1, x2, y2, x3, y3))
                                    ++ans;
    assert(ans % 6 == 0);
    cout << ans / 6 << '\n';
    return 0;
}
```

Задача VI.1.1.3. Голосовой помощник (20 баллов)

Условие

В современных городах очень легко заблудиться без навигатора. Развязки, ремонты, стройки — все это надо учитывать при выборе маршрута. Вы работаете в компании, которая поддерживает текущее состояние дорог и тротуаров. На сервере хранится карта города, разбитая на маленькие прямоугольники. Про каждый из них известно, есть через него проход или нет.

Вам поручили разработать логику голосового помощника, который будет строить маршрут по данным карты и возвращать текст для озвучки пользователю.

Формат входных данных

В первой строке даны 2 числа — N и M, про которые известно, что ($1 \leqslant N \leqslant 100$), ($1 \leqslant M \leqslant 100$).

Затем вводится N строк, в каждой из которых M символов, описывающих возможность прохода через ту или иную клетку. Каждый символ является либо точкой («.»), обозначающей наличие прохода, либо решеткой («#»), обозначающий преграду. Отдельными символами обозначается стартовая клетка («S») и финишная («F»). Необходимо проложить и озвучить кратчайший маршрут из точки S в точку F. длиной маршрута считается число шагов. Исходим из того, что повороты происходят моментально. Гарантируется, что на карте есть ровно один символ S и ровно один символ F.

Формат выходных данных

Выведите не более 1000 действий следующего вида:

«MOVE RIGHT» — поворот направо,

«MOVE LEFT» — поворот налево,

«GO» — движение прямо,

«FINISH» — сообщение пользователю о том, что он добрался до точки назначения.

Изначально вы смотрите вниз. Если первым шагом сделать GO, пользователь опустится на строку ниже в лабиринте.

Поле по периметру окружено решетками, т. е. случайно выйти за пределы поля не получится.

Если во время движения вы дойдете до стены и попытаетесь продолжить движение в стену, то получите вердикт «Wrong answer».

Правильным считается любой маршрут из начальной точки в конечную, использующий минимально возможное количество команд GO. Всего в маршруте должно быть не более 1000 команд.

Примеры

Пример №1

Стандартный ввод
5 6
######
#S...#
##.#.#
###F.#
######

Стандартный вывод
MOVE LEFT
GO
GO
GO
MOVE RIGHT
GO
GO
MOVE RIGHT
GO
FINISH

Решение

Воспользуемся поиском в ширину.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
n, m = map(int, input().split())

pole = []
for i in range(n):
    pole.append(input().strip())

start_x = -1
start_y = -1
for i in range(n):
    for j in range(m):
        if pole[i][j] == "S":
            start_x = i
            start_y = j

finish_x = -1
finish_y = -1
for i in range(n):
    for j in range(m):
        if pole[i][j] == "F":
            finish_x = i
            finish_y = j

dist = []
```

```
from1 = []
for i in range(n):
    ln = []
    from_line = []
    for j in range(m):
        ln.append(100000)
        from_line.append([-1, -1])
    dist.append(ln)
    from1.append(from_line)

q = [[start_x, start_y]]
dist[start_x][start_y] = 0
add = [[0, 1], [0, -1], [1, 0], [-1, 0]]
while len(q):
    now_x = q[0][0]
    now_y = q[0][1]
    q.pop(0)
    for add_x_y in add:
        new_x = now_x + add_x_y[0]
        new_y = now_y + add_x_y[1]
        if pole[new_x][new_y] != "#":
            if dist[new_x][new_y] > dist[now_x][now_y] + 1:
                q.append([new_x, new_y])
                dist[new_x][new_y] = dist[now_x][now_y] + 1
                from1[new_x][new_y] = [now_x, now_y]
path = [[finish_x, finish_y]]
while path[-1][0] != start_x or path[-1][1] != start_y:
    prev = from1[path[-1][0]][path[-1][1]]
    path.append(prev)
path.reverse()

napr = 0
add = [[1, 0], [0, -1], [-1, 0], [0, 1]]
ans = []
for i in range(1, len(path)):
    nw = path[i]
    pr = path[i - 1]
    dx = nw[0] - pr[0]
    dy = nw[1] - pr[1]

    mv = [dx, dy]
    if mv == add[(napr + 1) % 4]:
        ans.append("MOVE RIGHT")
        napr = (napr + 1) % 4
    else:
        while mv != add[napr]:
            ans.append("MOVE LEFT")
            napr = (napr + 3) % 4
    ans.append("GO")
ans.append("FINISH")

print('\n'.join(ans))
```

Задача VI.1.1.4. Работа с вероятностью (25 баллов)

Условие

В моделях машинного обучения для характеристик различных сущностей применяют теорию вероятности и многомерные векторы.

Например {1, 2} — двумерный вектор, у которого число 1 — первая корданата, {4, 3, 2, 5, 4} — пятимерный вектор, первая координата которо равняется 4.

Вы работете с некой сложной моделью ИИ, в которой координаты векторов — случайные целые числа из интервала $[0, K]$. K — некое целое число, которое заранее известно. Разработчики алгоритма попросили написать программу, которая вычислит важный коэффициент, необходимый для работы нейросети.

Его значение определяется как вероятность события, при котором среднее арифметическое первых координат N векторов будет равно дроби вида M/N, где M — некое число, значение которого вы знаете. Выведите ответ ровно с 6 знаками после запятой.

Напишите программу, которая рассчитает его значение.

Формат входных данных

В единственной строке даны 3 числа через пробел — N, M, K, про которые известно, что $(2 \leqslant N \leqslant 100), (0 \leqslant M \leqslant 1000), (1 \leqslant k \leqslant 100)$.

Формат выходных данных

Выведите единственное число — значение коэффициента ровно с 6 знаками после запятой.

Примеры

Пример №1

Стандартный ввод
`2 2 2`

Стандартный вывод
`0.333333`

Пример №2

Стандартный ввод
`3 5 4`

Стандартный вывод
`0.144000`

Пояснения к примеру

Пример №1

У первого вектора координата — случайное число из набора $\{0, 1, 2\}$. У второго также. Итого есть 9 случаев — $\{0, 0\}$, $\{0, 1\}$, $\{0, 2\}$, $\{1, 0\}$, $\{1, 1\}$, $\{1, 2\}$, $\{2, 0\}$, $\{2, 1\}$, $\{2, 2\}$. Вероятность каждого случая — $\frac{1}{9}$. Всего у нас 3 случая, когда сумма равняется двум — $\{0, 2\}$, $\{1, 1\}$, $\{2, 0\}$. Поэтому искомая вероятность равняется $\frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$.

Пример №2

Координата — случайное число из набора $\{0, 1, 2, 3, 4\}$. Всего есть 125 вариантов — $\{0, 0, 0\}$, $\{0, 0, 1\}$, ..., $\{4, 4, 3\}$, $\{4, 4, 4\}$. Вероятность каждого случая — $\frac{1}{125}$. Из них 18 имеют сумму 5 — $\{0, 1, 4\}$, $\{0, 2, 3\}$, $\{0, 3, 2\}$, $\{0, 4, 1\}$, $\{1, 0, 4\}$, $\{1, 1, 3\}$, $\{1, 2, 2\}$, $\{1, 3, 1\}$, $\{1, 4, 0\}$, $\{2, 0, 3\}$, $\{2, 1, 2\}$, $\{2, 2, 1\}$, $\{2, 3, 0\}$, $\{3, 0, 2\}$, $\{3, 1, 1\}$, $\{3, 2, 0\}$, $\{4, 0, 1\}$, $\{4, 1, 0\}$. Поэтому искомая вероятность равняется $\frac{18}{125}$.

Решение

Применим метод динамического программирования — посчитаем вероятность выпадения суммы j из первых i векторов.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;

using ld = long double;

int main() {
    size_t n, m, k; cin >> n >> m >> k;
    ld factor = (ld)1.0 / (ld)(k + 1);

    vector<vector<ld>> dp(n + 1, vector<ld>(m + 1));
    dp[0][0] = 1;

    for (size_t i = 1; i <= n; ++i) {
        for (size_t j = 0; j <= m; ++j) {

            for (int x = 0; x <= k && x <= j; ++x) {
                dp[i][j] += dp[i - 1][j - x] * factor;
            }
        }
    }

    cout << fixed << showpoint << setprecision(6) << dp[n][m] << endl;
}
```

Задача VI.1.1.5. Подбор музыки (30 баллов)

Условие

Одной из сфер применения ИИ является подбор музыки в музыкальных сервисах.

В данной задаче известна мелодия и музыкальные фрагменты, которые нравятся пользователю. Мелодия представляет собой числовую последовательность. Фрагменты — также числовые последовательности. Если фрагмент встречается единожды, он приносит 1^2 единицу удовольствия, дважды — 2^2 удовольствия. .... N раз встречающийся фрагмент приносит N^2 единиц удовольствия. Посчитайте, сколько единиц удовольствия принесет данная песня конкретному пользователю.

Формат входных данных

В первой строке дано 2 числа N, M – размер последовательности и число музыкальных фрагментов, $(1 \leqslant N \leqslant 3 \cdot 10^5), (1 \leqslant M \leqslant 10^5)$.

В следующей строке задано N чисел B_i через пробел — частоты проигрываемой мелодии $(1 \leqslant B_i \leqslant 50)$. Частоты усреднены, чтобы повысить шанс совпадения мелодии и вкусов пользователя.

Следующие M строк описывают фрагменты, которые нравятся пользователю. В начале каждой строки записано число $S_i, (1 \leqslant S_i \leqslant 10^5)$ — длина очередного фрагмента. Затем в этой же строке идут S_i чисел — частоты фрагмента $T_{ij}, (1 \leqslant T_{ij} \leqslant 50)$. Известно, что в базе на одного пользователя отводится не очень много места, из-за чего суммарная длина всех фрагментов также не превышает $3 \cdot 10^5$ символов. Формально говоря $S_1 + S_2 + ... + S_N \leqslant 3 \cdot 10^5$.

Один и тот же фрагмент может встретиться несколько раз.

Формат выходных данных

Выведите одно число — сколько единиц удовольствия принесет данная песня конкретному пользователю.

Примеры

Пример №1

Стандартный ввод
10 5 2 9 24 28 24 21 7 11 24 21 3 27 1 39 3 37 4 47 3 20 5 18 4 24 21 7 11 4 2 9 24 28

Стандартный вывод
2

Пояснения к примеру

Начало мелодии (первые 4 частоты) [2, 9, 24, 28] совпадают с фрагментом 5.

Также отрезок мелодии с 5-го по 8-й ноту [24, 21, 7, 11] совпадает с фрагментом 4.

Можно убедиться, что других полных совпадений фрагментов с мелодией нет.

Решение

Давайте разделим строки на легкие и тяжелые. Легкими будем считать те, длина которых меньше $\sqrt{M}$.

Длинными — те, длина которых более $\sqrt{M}$. Первым шагом разобьем строки на 2 массива, и решим для каждых задачу разными способами.

Решим задачу для длинных строк. Сколько их может быть? Если длина хотя бы $\sqrt{M}$, а суммарная длина — M, то их количество меньше или равно $\sqrt{M}$. Воспользуемся префикс-функцией для поиска всех вхождений строки в текст для каждой строки — $O(\text{длина строки} + N)$. Итого решение длинных строк — $O(M + \sqrt{M} \cdot N)$.

Для решения коротких строк воспользуемся хешированием. Посчитаем хеши всех префиксов. Используя их, выпишем хеши всех подстрок длины 1, 2, ..., $\sqrt{M}$ в словарь так, чтобы ключом было значением хеша, значенем ключа — сколько раз ключ встречался в строке. Затем считаем хеш каждого фрагмента и из словаря, соответствующего длине фрагмента, узнаем число вхождений в мелодию. Итого решение за $O(N \cdot \sqrt{M} + M)$, где первое слагаемое — подсчет хешей отрезков, M — суммарное время вычисления хешей фрагментов. Сложив ответы для длинных и коротких строк, получим ответ на задачу. Итоговая сложность решения — $O(N \cdot \sqrt{M} + M)$.

Такой подход к решению задач — отдельно разбираться с большими, отдельно с маленьким называется корневой декомпозицией.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <bits/stdc++.h>

using namespace std;

vector<int> prefix_function(const vector<int>& s) {
    int n = (int)s.size();
    vector<int> pi(n);
    for (int i = 1; i < n; i++) {
        int j = pi[i-1];
        while (j > 0 && s[i] != s[j])
            j = pi[j-1];
        if (s[i] == s[j])
            j++;
        pi[i] = j;
    }
    return move(pi);
}

int countIn(int sz, const vector<int>& s) {
    vector<int> ans = prefix_function(s);
```

```
    int cnt = 0;
    for (auto x : ans)
        if (x == sz)
            ++cnt;
    return cnt * cnt;
}

long long long_solve(vector<vector<int>>&& long_parts, const vector<int>& text) {
    long long ans = 0;
    for (const auto& line : long_parts) {
        vector<int> s;
        s.reserve(line.size() + 1 + text.size());
        for (auto x : line)
            s.push_back(x);
        s.push_back(-1);
        for (auto x : text)
            s.push_back(x);
        ans += (long long)countIn(line.size(), s);
    }
    return ans;
}

const int MAXN = 300000;
const long long P = 53;
const long long MOD = 1e9 + 7;
long long p[MAXN], pref_hash[MAXN];

void precalc() {
    p[0] = 1;
    for (int i = 1; i < MAXN; ++i)
        p[i] = p[i - 1] * P % MOD;
}

void calc_text_hash(const vector<int>& text) {
    pref_hash[0] = 0;
    for (int i = 0; i < text.size(); ++i) {
        pref_hash[i + 1] = (pref_hash[i] * P + text[i]) % MOD;
    }
}

long long get_hash(int l, int r) {
    return (pref_hash[r] + (MOD - pref_hash[l]) * p[r - l]) % MOD;
}

void recount(const vector<int>& text, unordered_map<int, int>& hashs, int len) {
    hashs.clear();
    for (int i = 0; i + len <= text.size(); ++i)
        ++hashs[get_hash(i, i + len)];
}

int count_hash(const vector<int>& line) {
    long long hash = 0;
    for (auto el : line)
            hash = (hash * P + el) % MOD;
    return (int)hash;
}

long long short_solve(vector<vector<int>>&& short_parts, const vector<int>& text) {
    sort(short_parts.begin(), short_parts.end(),[] (const vector<int>& lhs, const vector<int>& rhs) {return lhs.size() < rhs.size();});
```

```cpp
    long long ans = 0;
    int len_counted = 0;
    unordered_map<int, int> cnt_by_hash;
    for (const auto& line : short_parts) {
        if (line.size() != len_counted) {
            recount(text, cnt_by_hash, line.size());
            len_counted = line.size();
        }
        long long cnt_in = (long long)cnt_by_hash[count_hash(line)];
        ans += cnt_in * cnt_in;
    }
    return ans;
}

int main() {
    precalc();
    cin.tie(nullptr);
    cout.tie(nullptr);
    ios_base::sync_with_stdio(false);
    int n, m;
    cin >> n >> m;
    vector<int> text(n);
    for (int i = 0; i < n; ++i)
        cin >> text[i];
    calc_text_hash(text);
    vector<vector<int>> long_parts, short_parts;
    const int MID = 300;
    for (int i = 0; i < m; ++i) {
        int sz_line;
        cin >> sz_line;
        (sz_line > MID) ? long_parts.push_back({}) : short_parts.push_back({});
        vector<int>& new_line = (sz_line > MID) ? long_parts.back() : short_parts.back();
        new_line.resize(sz_line);
        for (int i = 0; i < sz_line; ++i)
            cin >> new_line[i];
    }
    long long ans = long_solve(move(long_parts), text) + short_solve(move(short_parts), text);
    cout << ans << '\n';
    return 0;
}
```

Тестовые наборы для задач представленны по ссылке — https://disk.yandex.ru/d/-LVERknoaUJZCA.

Математика. 8–9 классы

Задача VI.1.2.1. (15 баллов)

Условие

Момент времени будем называть *красивым*, если после выписывания даты в формате «год.месяц.день» и приписывания далее точного времени в 24-часовом формате получается симметричная последовательность цифр, то есть такая, которая слева направо читается так же, как и справа налево. Например, 2020.01.10.02:02. Определите последний в этом тысячелетии красивый момент.

Решение

Так как момент времени должен быть в этом тысячелетии, то первая (а значит, и последняя) цифра равна 2. Отметим, что всего цифр в записи тогда 12, из которых две уже установлены: $2{*}{*}{*}.{*}{*}.{*}{*}.{*}{*}:{*}2$. Будем называть искомый в задаче момент *моментом X*.

Предпоследняя цифра является разрядом десятков в минутах, то есть не может быть больше 5. Тогда симметричная ей цифра в разряде сотен лет момента X не может быть более 5 (а 5 может). Максимальная цифра в следующем разряде — это 9, значит число на первых трёх позициях записи момента не может быть более 259, что соответствует последним трём цифрам $9:52$. Предшестующая им четвёртая с конца цифра представляет разряд десятков в часе момента X. То есть эта цифра может быть 0, 1 или 2. При этом для следующей цифры 9 (в третьем с конца разряде) текущая цифра не может быть 2 (так как количество часов в такой записи не может быть более 23). Значит, максимальная цифра на четвёртой позиции (при условии, что на предыдущих 259) это 1.

Пятая и шестая цифры в записи — месяц момента X. Не может быть более 12. Тогда самое позднее начало из перых шести цифр — это 2591.12. Оно полностью и однозначно определяет весь момент, так как цифры с 7-ой по 12-ю — это зеркальное отражение уже найденных цифр.

Ответ: 2591.12.21.19:52.

Примечание: в случае трактовки тысячелетия от 2001 до 3001 года решение аналогичное и ответ 3000.12.21.00:03.

Критерии оценивания

- Только ответ — 5 баллов.
- Ошибка в рассуждении про один из разрядов (не учитывая симметричный разряд) — не более 10 баллов.
- Ошибки в рассуждениях по поводу двух несимметричных разрядов — не более 5 баллов.

Задача VI.1.2.2. (15 баллов)

Условие

Петя, Вика, Толя, Чулпан, Потап, Шарлотта стоят в указанном порядке по кругу, считая по часовой стрелке, и смотрят в центр круга (то есть по правую руку от Пети стоит Шарлотта, а по левую — Вика). Каждый мальчик загадал число и сообщил его шёпотом обеим соседкам. Каждая девочка умножила число от своего соседа против часовой стрелки на 2, и к этому результату прибавила число от соседа по часовой стрелке. В результате у Вики получилось число 57, у Чулпан — число 53, а у Шарлотты — число 31. Какое число загадал Петя?

Решение

Обозначим число Пети за x, число Толи за y, число Потапа за z. Тогда число Вики равно $y + 2x$, оно же равно 57. Аналогично, число Чулпан равно $z + 2y = 53$, число Шарлотты равно $x + 2z = 31$. Сложим все эти три числа отдельно левые части и отдельно правые. Получим равенство $3x+3y+3z = 141$. То есть $x+y+z = 141/3 = 47$.

Пользуясь установленным равенством и равенством для числа Шарлотты, имеем $(x+2z)-(x+y+z) = 31-47$, то есть $z-y = -16$. Аналогично, пользуясь выражением для числа Чулпан, имеем: $y - x = 53 - 47 = 6$. Тогда можем выразить: $y = x + 6$, $z = y - 16 = x - 10$. Подставляя в выражение с суммой всех, имеем: $x + (x + 6) + (x - 10) =$
$= 47$, то есть $3x - 4 = 47$ и $x = 17$.

Ответ: 17.

Критерии оценивания

- Только ответ — 5 баллов.
- Ошибка, связанная с поворотом по или против часовой стрелки — не более 12 баллов (−3 балла).
- Логическая ошибка прм решении системы — не более 10 баллов.
- Две логические ошибки — не более 5 баллов.
- Арифметические ошибки, не влияющие на ход решения — −3 балла.
- Арифметические ошибки, существенно влияющие на ход решения (ведущие к другой линии решения и проч.) — не более 5 баллов.

Задача VI.1.2.3. (20 баллов)

Условие

Площадь трапеции $ABCD$ равна 164 см^2, длина её высоты равна 8 см, боковые стороны AB и CD равны 10 см и 17 см соответственно. Чему равна длина основания BC?

Решение

Опустим из точек A и B перпендикуляры на прямую CD, основания перпендикуляров, соответственно, P и Q. Тогда треугольники ABP и DCQ суть прямоугольные, в которых известны гипотенуза и катет (в обоих случаях один из катетов равен длине перпендикуляра между прямыми AD и BC, то есть 8). Тогда по теореме Пифагора имеем, что $AP = \sqrt{AB^2 - BP^2} = \sqrt{10^2 - 8^2} = 6$, аналогично $DQ = 15$.

Площадь прямоугольника $PBCQ$ равна площади всей трапеции за вычетом площадей треугольников ABP и CDQ. При этом одна сторона прямоугольника равна 8. Значит, другая сторона равна: $(164 - 8 \cdot 6/2 - 8 \cdot 15/2)/8 = (41 - 21)/2 = 10$.

Ответ: 10.

Критерии оценивания

- Только ответ — 5 баллов.
- Найдены площади прямоугольных треугольников отсекаемых высотами — +5 баллов.

Задача VI.1.2.4. (25 баллов)

Условие

Функция $P(x)$ определяется как $P(x) = x^2 - 38x - 80$.

Решите уравнение $P(x) = P\left(\dfrac{x+4}{x+1}\right)$.

Решение

Функция $P(x)$ — это квадратичная функция, график которой — это парабола «усами вверх». Тогда для значения y_0 либо не существует решений уравнения $P(x) = y_0$, либо решение одно, если y_0 соответствует вершине данной параболы (для $x = -b/2a$, то есть $38/2 = 19$ в данном случае), либо решения два и они симметричны относительно x-координаты вершины параболы (то есть относительно $-b/2a = 19$ в данной задаче). Причём второй случай принято считать частным случаем третьего случая, просто два корня равны друг другу.

Воспользуемся этим правилом. Тогда если $P(x) = P\left(\dfrac{x+4}{x+1}\right)$ для какого-то x, то получаем, что либо $x = \dfrac{x+4}{x+1}$, либо эти два значения симметричны относительно x-координаты вершины параболы, то есть относительно 19.

Первое условие превращается в квадратное уравнение $x(x+1) = x+4$ после домножения на $x+1$ (и замечания, что x не равно -1). Решения этого уравнения 2 и -2.

Второе условие (симметричности) записывается как $\left(x + \dfrac{x+4}{x+1}\right)/2 = 19$. После домножения на знаменатель также получается квадратное уравнение с корнями $18 - \sqrt{358}$ и $18 + \sqrt{358}$, что завершает доказательство.

Ответ: -2, 2, $18-\sqrt{358}$, $18+\sqrt{358}$.

Критерии оценивания

- Только ответ — 5 баллов.
- Упущены решения в ответе — -2 балла за каждый упущенный.
- Разобран только случай $x = \dfrac{x+4}{x+1}$ — $+7$ баллов.
- Неверные выводы про квадратичную функцию — -5 баллов.

Задача VI.1.2.5. (25 баллов)

Условие

Сферическим кодом будем называть несколько точек в четырёхмерном пространстве (то есть с четырьмя координатами), каждая из координат которых равна либо 0, либо 1, при условии, что все попарные расстояния между этими точками одинаковы. Какое максимальное количество точек может быть в сферическом коде?

Напоминание: расстояние между двумя точками 4-мерного пространства вычисляется по четырёхмерной теореме Пифагора: расстояние между (a, b, c, d) *и* (x, y, z, t) *равно* $\sqrt{(a-x)^2+(b-y)^2+(c-z)^2+(d-t)^2}$.

Решение

Приведём пример сферического кода с 4-мя точками: $(0,0,0,0)$; $(1,1,0,0)$; $(1,0,1,0)$; $(1,0,0,1)$. Отметим, что любые две точки имеют две совпадающие координаты и две отличающиеся, то есть расстояние между любыми двумя из них равно $\sqrt{0^2+0^2+1^2+1^2} = \sqrt{2}$, то есть это — сферический код.

Покажем, что пять и более точек в сферическом коде (для четырёх координат) быть не может. Предположим противное, то есть существуют 5 точек, образующих точки сферического кода. Тогда во-первых, все эти точки имеют одинаковую чётность суммы координат (то есть для каждой точки рассматриваем сумму её координат). Действительно: раз расстояния между каждой парой точек равны, то и квадраты расстояний равны, а это целые числа. Тогда для любых трёх точек из сферического кода имеем следующие сравнения по модулю 2 (исходя из того, что для произвольного n верно $n^2 \equiv n \mod 2$): пусть рассматриваются точки (a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4) и (c_1, c_2, c_3, c_4). Тогда имеем:

$$(a_1-b_1)+(a_2-b_2)+(a_3-b_3)+(a_4-b_4) \equiv (a_1-b_1)^2+(a_2-b_2)^2+(a_3-b_3)^2+(a_4-b_4)^2 \equiv$$

$$\equiv (c_1-b_1)^2+(c_2-b_2)^2+(c_3-b_3)^2+(c_4-b_4)^2 \equiv (c_1-b_1)+(c_2-b_2)+(c_3-b_3)+(c_4-b_4)$$

(все сравнения указаны по модулю 2), откуда из первого и последнего равенств в цепочке имеем:

$$a_1+a_2+a_3+a_4 \equiv c_1+c_2+c_3+c_4 \mod 2.$$

Всего количество точек с четырьмя координатами 0 и 1 равно $2^4 = 16$. Тех, у кого сумма координат чётная, столько же, сколько тех, у кого сумма координат нечётная, а именно 8. При этом если в набор одновременно попали две точки, у которых в каждой из координат стоят разные числа (в одном 0, а в другом 1), то расстояние между такими точками равно $\sqrt{4} = 2$, что возможно только в подобной паре «дополнительных» друг для друга точек. До любой из других точек от любой из этих двух расстояние будет не более $\sqrt{3}$, что меньше 2. То есть если в сферическом коде есть две дополнительные друг другу точки, то других точек в этом сферическом коде нет. Но мы уже знаем, что есть сферический код, в котором больше 2-х точек.

Таким образом, если в сферическом коде более 2-х точек, то в нём нет дополнительных друг другу точек. Но все пары точек с четырьмя координатами, каждая из которых 0 или 1, разбиваются на пары дополнительных, причём суммы координат обеих точек будут иметь одну чётность. Значит, среди точек сферического кода будет не более половины от общего количества точек с суммой координат такой чётности. То есть их не более $8/2 = 4$.

Ответ: 4.

Критерии оценивания

- Ответ — 5 баллов.
- Пример — 5 баллов.
- Доказано, что точки одной чётности — +5 баллов.
- Доказано, что если в наборе есть пара дополнительных друг другу точек, то других точек в наборе нет — +3 балла.

Математика. 10–11 классы

Задача VI.1.3.1. (15 баллов)

Условие

Определите количество решений уравнения $\cos^2 x + (x - 1)\sin x = \sin x + 1$ на отрезке $[-\pi, \pi]$.

Решение

Перенесём все слагаемые в левую часть выражения, применим основное тригонометрическое тождество $\cos^2 x + \sin^2 x = 1$. Получим уравнение $-\sin^2 x + (x-2)\sin x = 0$, то есть $\sin x(-\sin x + x - 2) = 0$. Это равенство выполнено, когда нулю равен либо первый множитель ($\sin x$), либо второй множитель ($-\sin x + x - 2$), либо оба одновременно. Первый множитель равен нулю на интервале $[-\pi, \pi]$ в трёх точках: $-\pi, 0$ и π. В других точках нулю он не равен.

Условие равенства второго множителя нулю перепишем в виде $\sin x = x - 2$. Левая часть выражения при всех значениях x принимает значения не менее -1 (так как это стандартная синусоида), а правая часть выражения строго меньше -1 при x меньше или равных 0 (так как это возрастающая линейная функция, которая равна -2 при $x = 0$). Поэтому на интервале $(-\pi, 0]$ рассматриваемое равенство невозможно.

Далее на интервале $(0, 2]$ выражение $\sin x$ принимает положительные значения (так как $2 < \pi$), а выражение $x - 2$ на этом интервале неположительно. Значит, на этом интервале равенство $\sin x = x - 2$ также невозможно.

На интервале $(2, \pi)$ выражение $\sin x - (x - 2)$ имеет одно решение, так как эта разность непрерывна, строго убывает на данном интервале $\sin x$ строго убывает, $-(x-2)$ также строго убывает, а значения в концах интервала имеют разные знаки ($\sin 2 - (2 - 2) = \sin 2 > \sin \pi = 0$; $\sin \pi - (\pi - 2) = 2 - \pi < 0$).

Ответ: 4.

Критерии оценивания

- Только ответ — 5 баллов.
- В целом верно, но из-за арифметических ошибок неверный ответ — не более 12 баллов.
- Логические ошибки, либо более трёх арифметических ошибок — не более 10 баллов.

Задача VI.1.3.2. (15 баллов)

Условие

Каждый день перед сном Коля смотрит юмористические шоу с кружечкой любимого напитка: чай, кофе или какао. Для этого каждый раз он случайно и равновероятно выбирает напиток. Найдите вероятность, что за неделю он выпьет каждого любимого напитка хотя бы один раз.

Решение

Исходя из условий задачи, каждая из последовательностей из 7-ми выборов напитка равновероятна. Поэтому для нахождения искомой вероятности вычислим общее количество исходов и число подходящих исходов. Не подходят последовательности, в которых встречаются только два или только один напиток. Последовательностей, в которых встречаются только чай и кофе, как легко видеть, равно 2^7. Однако при таком подсчёте по каждой из пар окажется, что последовательности ровно с одним напитком будут посчитаны дважды. Тогда общее количество благоприятных исходов равно $3^7 - 3 \cdot 2^7 + 3$, а искомая вероятность $\dfrac{3^7 - 3 \cdot 2^7 + 3}{3^7}$.

Ответ: $\dfrac{602}{729}$.

Критерии оценивания

- Только верный ответ — 5 баллов.
- Не работающие в этой задаче, но верные в целом схемы вычисления — не более 5 баллов.

Задача VI.1.3.3. (20 баллов)

Условие

Дан равносторонний треугольник PQR. Внутри угла PRQ взята точка T. Найдите $\angle PRT$, если $\angle PTR = 20^\circ$ и $\angle QTR = 30^\circ$.

Решение

Рассмотрим поворот треугольника RTQ вокруг R на 60° (в том же направлении, поворот от луча RQ до луча RT). Пусть точка T при этом повороте перейдёт в некоторую точку S. Тогда треугольник TRS равносторонний (так как $RT = RS$ и $\angle TRS = 60^\circ$ по выбору поворота).

При указанном повороте точка Q перейдёт в точку P (так как $QR = PR$, $\angle QRP = 60^\circ$). Значит, треугольник RTQ при данном повороте переходит в треугольник RSP. Тогда верно $\angle RSP = \angle RTQ = 30^\circ$. Но тогда SP — биссектриса равностроннего треугольника RST: $\angle RSP = \angle PST$.

Из симметрии правильного треугольника RST относительно своей биссектрисы SP имеем, что $\angle SRP = \angle STP$. $\angle STP = \angle STR - \angle PTR = 60^\circ - 20^\circ = 40^\circ$. Тогда искомый в задаче угол $\angle PRT = \angle SRT - \angle SRP = 60^\circ - 40^\circ = 20^\circ$.

Ответ: 20°.

Критерии оценивания

- Только ответ — 5 баллов.
- Найдены другие углы, из которых всё ещё неясно, как подобраться к требуемому углу — +0 баллов (не оценивается).
- Есть идея поворота на 60° — +3 балла.
- Указано, что в результате такого поворота получается р/с треугольник(и) — +2 балла (суммируется с предыдущим).
- Нетривиальные продвижения, которые, тем не менее, неясно, как помогут в задаче — не более +3 балла.

Задача VI.1.3.4. (25 баллов)

Условие

Произведением двух матриц размера 2×2, записанных в виде $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ и $\begin{pmatrix} x & y \\ z & t \end{pmatrix}$, называется матрица $\begin{pmatrix} ax+bz & ay+bt \\ cx+dz & cy+dt \end{pmatrix}$. k-ой степенью квадратной матрицы M (размера $m \times m$) называется выражение $M \cdot M \cdot \ldots \cdot M$ с k множителями (умножения в такой ситуации выполняются по очереди). Найдите 20-ю степень матрицы $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.

Решение

Обозначим $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$ как M. Докажем, что M^k имеет вид $\begin{pmatrix}F_{k+1} & F_k\\ F_k & F_{k-1}\end{pmatrix}$, где F_k – числа Фибоначчи, то есть последовательность, заданная условиями $F_1 = F_2 = 1$ и $F_{i+2} = F_i + F_{i+1}$. Доказательство проведём по индукции: база $k = 1$ уже фактически проверена. Пусть утверждение доказано для всех t от 1 до m. Тогда для $M^{m+1} = M^m \cdot E_0 = \begin{pmatrix}F_{m+1} & F_m\\ F_m & F_{m-1}\end{pmatrix} \cdot \begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix} = \begin{pmatrix}F_{m+1} + F_m & F_{m+1}\\ F_m + F_{m-1} & 0\end{pmatrix}$, что и даёт требуемое.

Остаётся вычислить числа Фибоначчи до соответствующего номера. Для матрицы M^{20} необходимыми будут элементы последовательности с номерами $21, 20$ и 19. Это можно делать многими способами, например, заполняя соответствующую таблицу. Выпишем числа Фибоначчи пятёрками: $1, 1, 2, 3, 5, \quad 8, 13, 21, 34, 55, \quad 89, 144, 233, 377, 610, \quad 987, 1597, 2584, 4181, 6765, \quad 10946, \ldots$.

Таким образом, ответ в задаче: $\begin{pmatrix}10946 & 6765\\ 6765 & 4181\end{pmatrix}$.

Критерии оценивания

- Только ответ — 5 баллов.
- Задача сформулирована в терминах чисел Фибоначчи — +5 баллов.
- Найдена закономерность про матрицы и числа Фибоначчи, либо что-то аналогичное — +5 баллов.

Задача VI.1.3.5. (25 баллов)

Условие

Клетки клетчатой полосы $n \times 1$ изначально покрашены в два цвета. Разрешается выбрать любую клетку вместе с соседними с ней по сторонам, и перекрасить в противоположный цвет каждую из выбранных клеток. Сколько различных покрасок этой полосы можно получить таким образом?

Решение

Проверим, что при $n = 1$ число способов (которое обозначим за Q) равно 2 (первый либо второй цвет). При $n = 2$ число способов Q равно также 2, так как при выборе любой клетки вместе с ней окажется выбрана и вторая клетка. При $n = 3$ имеем такую ситуацию, что при выборе центральной клетки выбранными оказываются все три клетки полосы. При выборе угловой клетки выбраны оказываются две крайние клетки. Отсюда видно, что за два перекрашивания можно добиться того, чтобы перекрасить только угловую клетку (любую из двух). Но тогда и одну центральную клетку перекрасить не составит труда. Значит, каждую из клеток мы научились перекрашивать произвольно и независимо от других. Значит, итоговый ответ при $n = 3$: $Q = 8 = 2^3$.

«Переведём» задачу на язык векторов с координатами 0 и 1: будем обозначать клетку первого цвета на полосе цифрой 1, клетку второго цвета — цифрой 0. Далее каждому перекрашиванию сопоставим вектор из n нулей и единиц, соответствующий перекрашиваемым клеткам, в котором цифра 1 означает перекрашивание соответствующей клетки, а цифра 0 — сохранение без изменений. Тогда легко видеть, что итог перекрашивания обозначается вектором длины n (всё с тем же соответствием, что 1 обозначает первый цвет, а 0 — второй), равным сумме исходного вектора цветов и вектора перекрашивания, взятому по модулю 2 (по каждой координате независимо).

Для i-ой клетки полосы обозначим соответствующий ей вектор перекрашивания, который в случае $i = 1$ или $i = n$ содержит только две единицы (считаем, что номера координат идут слева направо, то есть в векторе $(1, 1, 0, \dots, 0)$ первая и вторая координаты равны 1). Поэтому, это вектора $(1, 1, 0, \dots, 0)$ и $(0, \dots, 0, 1, 1)$ соответственно. Иначе i-ой клетке соответствует вектор $(0, \dots, 0, 1, 1, 1, 0, \dots, 0)$, в котором единицы на $i - 1$-ой, i-ой и $i + 1$-ой позициях.

Утверждение 1. Итоговый результат набора нескольких перекрашиваний не зависит от порядка операций.

Доказательство: утверждение очевидно, так как из сказанного выше следует, что вектор итогового узора равен вектору начального узора плюс сумма векторов перекрашиваний, и всё это по модулю 2 (по каждой координате независимо). Такое выражение не зависит от порядка слагаемых.

Утверждение 2. Итоговое количество возможных различных узоров не зависит от начальной покраски.

Доказательство: прибавив ко всем возможным итоговым узорам вектор начальной покраски (по модулю два покоординатно), получим набор из того же количества различных векторов. При этом легко видеть, что полученные вектора будут соответствовать тем же перекрашиваниям, что и ранее, но только относительно начальной покраски со всеми клетками второго цвета (то есть с вектором покраски, состоящим из всех нулей).

Таким образом, далее будем считать, что изначально полоса покрашена только во второй цвет. На ответ это не повлияет.

Утверждение 3. Для произвольного набора перекрашиваний найдётся набор с тем же итоговым узором, но в котором не будут присутствовать «дубли» — одинаковые перекрашивания два и более раз.

Доказательство: для доказательства достаточно сгруппировать повторяющиеся перекрашивания подряд (согласно утверждению 1 итоговый узор не поменяется), а затем исключать пары одинаковых векторов перекрашиваний, пока такие пары есть в наборе. Очевидно, при удалении любой такой пары итоговый узор не изменится, так как сумма векторов двух одинаковых перекрашиваний равна 0 (по модулю 2).

Таким образом, далее мы будем искать только количество узоров для наборов неповторяющихся перекрашиваний.

Утверждение 4. Если два различных набора перекрашиваний (каждый набор без дублей, наборы отличаются не только порядком входящих в них перекрашиваний) дают одинаковый итоговый узор, то после объединения таких наборов и избавления от дублей, получится непустой набор, дающий нулевой итоговый вектор перекрашивания.

Доказательство: просто рассмотрим этот объединённый набор. С одной стороны, так как итоговый узор равен сумме векторов перекрашиваний, то итоговый вектор перекрашивания объединённого набора — это сумма двух одинаковых векторов по модулю 2. Но это нулевой вектор. Кроме того, в каждом из наборов нет дублей векторов. Это значит, что в объединённом наборе дубли берутся только из векторов, которые были представлены в единственном экзмепляре в обоих наборах. Но тогда если все вектора разбились на дубли, это значит, что в двух наборах содержатся одинаковые наборы векторов, что противоречит предположению. Утверждение доказано.

Утверждение 5. Пусть набор векторов перекрашиваний не пуст, не содержит дублей и в сумме даёт нулевое итоговое изменение. Тогда n сравнимо с 2 по модулю 3, и в набор входят все вектора перекрашиваний, соответствующие клеткам полосы с номером, не делящимся на 3.

Доказательство: рассмотрим произвольный такой непустой набор перекрашиваний без дублей, который в результате даёт нулевое изменение. Рассмотрим минимальный номер i клетки, для которой вектор перекрашивания входит в набор. Если i не равно 1, то, очевидно, $i-1$-ая клетка будет перекрашена, причём ровно 1 раз. Но значит, в результате эта клетка будет покрашена не как изначально — противоречие с условием. Значит, минимальный номер i равен 1. Следовательно, вектор перекрашивания для клетки 1 входит в набор. Так как эта клетка не должна поменять цвет в итоге, то также в набор должен входить вектор 2. Аналогично имеем, что вектор 3 не входит в набор. Далее аналогично легко установить, что вектора 4 и 5 должны присутствовать, а вектор 6 должен отсутствовать и так далее — все вектора с номерами кратными 3 должны отсутствовать, а остальные — присутствовать.

Остаётся заметить, что эти рассуждения верны и для другого конца полосы. Таким образом, вектора с номерами n и $n-1$ должны присутствовать в наборе, откуда имеем, что это клетки с номерами вида $3k+1$ и $3k+2$. В частности, получаем, что $n = 3k+2$, то есть сравним с 2 по модулю 3.

Тогда получаем, что в случае, когда n несравним с 2 по модулю 3, любой набор перекрашиваний сводится к набору перекрашиваний без дублей (с сохранением итогового узора), а все такие наборы дают различные результаты (иначе, если бы нашли два различных, дающих одинаковый результат, их объединение и удаление дублей дало бы непустой набор перекрашиваний с нулевым результатом (утверждение 4), но такого набора при n, несравнимом с 2, не существует (утверждение 5)). Значит, различных итоговых узоров столько же, сколько наборов перекрашиваний (без дублей) — это 2^n.

Если же n сравнимо с 2 по модулю 3, то аналогично любой набор сводится к набору без дублей, но есть единственный невырожденный «нулевой» набор (то есть дающего нулевой результат). При помощи него перекрашивание, задаваемое крайне правой клеткой, можно заменить на комбинацию нескольких других перекрашиваний. Но значит так, как перекрашивания задаются своими векторами при помощи суммы по модулю 2, то получается, это перекрашивание (задаваемое вектором номер n) можно заменить на эту комбинацию внутри любого набора из перекрашиваний не изменим итоговый результат (и отбросив дубли, если такие появились). Значит, все узоры, которые можно получить в этом случае, можно получить и при помощи первых $n-1$ векторов. Причём среди таких комбинаций уже не может быть пары различных наборов с одинаковым итоговым узором (иначе нулевая комбинация векторов должна была бы существовать и на первых $n-1$ координатах с нулевой последней координатой, но единственная нетривиальная нулевая комбинация имеет единицы на концах, как мы установили в утверждении 5 Таким образом, если n сравнимо с 2 по модулю 3, ответ равен 2^{n-1}.

Критерии оценивания

- Только ответ — 5 баллов.
- Переформулировано в терминах векторов — +2 балла.
- Каждое из сформулированных в доказательстве утверждений +2 балла (даже если не на языке векторов).
- Неправильный подсчёт финальных вариантов, когда уже доказано, что сводится к наборам без дублей — не более 15 баллов.

Инженерный тур

Общая информация

Задача предполагает создание алгоритма, способного ответить на вопросы к видео, заданные на естественном (русском) языке.

Актуальность задачи

Решение подобных задач способствует развитию технологий автоматического анализа визуальных данных. Автоматический анализ визуальных данных в значительной степени сокращает объем анализа по сравнению с ручной обработкой и создает возможности для автоматизации решения таких задач, как: поиск нужного фрагмента видео по запросу (например, на камерах видеонаблюдения), поиск в базе данных различных видео (например, классификация видео с определенными участниками) и так далее.

Требования к команде и компетенциям участников

Максимальное число участников в команде — 2. При этом мы не выделяем ролей и предлагаем участникам быть равноправными партнерами.

В качестве рекомендации участникам предлагалось при объединении выбирать партнеров таким образом, чтобы вместе они могли на высоком уровне закрыть большинство необходимых компетенций.

Оборудование и программное обеспечение

Участникам были предоставлены ноутбуки: оперативная память — 8 Гб, хранилище SSD — 256 Гб, процессор — Intel Core i5.

Предустановленное ПО:

- Браузер Chrome, Версия 108.0.5359.95.
- PyCharm, community edition, Версия 2021.3.2.
- Jupyter Notebook, из дистрибутива Anaconda 3 2020.11.
- Python 3.10.
- GIT 2.34.1.
- MS Office 2016.

Дополнительно для обучения моделей участникам был предоставлен доступ к сервису ML Space, где они получали возможность обучения моделей на 1 GPU Nvidia A100 в течение 20 ч.

Описание задачи

В задаче Video Question Answering модели необходимо проанализировать короткие видеофрагменты и представленные к ним вопросы по содержанию и запечатленным событиям и/или действиям, чтобы сгенерировать наиболее подходящие ответы на русском языке.

Участникам был предоставлен обучающий набор данных, который включал в себя 2700+ коротких видео и соответствующих им вопросов и ответов на русском языке, а также базовое решение от разработчиков с докером и примером загружаемого решения.

Так как участники не имели доступа к проверяющему датасету, в качестве решения на платформу нужно было загрузить обученную модель. Решения на проверяющей платформе запускались в изолированном окружении при помощи Docker. Время и ресурсы во время тестирования ограничены.

Система оценивания

Оценка решений производилась автоматически на основании метрики BLEU, которая позволяет сравнить эталонный и предсказанный текст. При этом BLEU оценивает не только соответствие отдельных слов, но и ;-грамм, содержащихся в тексте. Метрика BLEU была изначально предложена для оценки качества машинного перевода, однако она может применяться в любых задачах, в которых необходимо оценить близость двух текстов (при этом допуская вариативность текстов-кандидатов, что важно в задаче описания видео).

Проверочный набор данных был разделен на 2 части: публичную и приватную. Результаты участников, сформированные на основании публичной части, были доступны в ходе всего соревнования. После завершения приема решений был сформирован итоговый рейтинг на основании приватной части проверочного набора данных.

После формирования итогового рейтинга участникам будут начислены баллы, рассчитанные по формуле:

$$Total = \frac{Result \times (RankParticipants + 1 - RankPlace) \times 100}{MaxResult \times Participants},$$

где $Total$ — итоговый балл за решение задачи;

$Result$ — результат решения задачи;

$RankParticipants$ — общее количество участников в рейтинге;

$RankPlace$ — место участника в рейтинге;

$MaxResult$ — максимальный результат решения задачи среди всех участников;

$Participants$ — общее количество участников.

Участникам этапа, не загрузившим решение в сроки проведения соревнования, было начислено 0 баллов.

Решение задачи

Рассмотрим модель ответа на вопрос по видео на основе архитектуры CLIP PREFIX CAPTION, включающую в себя CLIP и ruGPT2small.

Установка библиотек

Устанавливаем библиотеки, под которыми запускается данное решение.

```
!pip install wandb
!pip install transformers
!pip install git+https://github.com/openai/CLIP.git
!pip install nltk
```

Инициализируем модель

```
import torch
import torch.nn as nn
from torch.nn import functional as nnf
from torch.utils.data import Dataset, DataLoader
from transformers import GPT2Tokenizer, GPT2LMHeadModel, AdamW,
↪  get_linear_schedule_with_warmup
from tqdm import tqdm, trange
import os
import pickle
import sys
import argparse
import json
from typing import Tuple, Optional, Union
from torch.cuda.amp import autocast

import clip

class MLP(nn.Module):
    def __init__(self, sizes: Tuple[int, ...], bias=True, act=nn.Tanh):
        super(MLP, self).__init__()
        layers = []
        for i in range(len(sizes) - 1):
            layers.append(nn.Linear(sizes[i], sizes[i + 1], bias=bias))
            if i < len(sizes) - 2:
                layers.append(act())
        self.model = nn.Sequential(*layers)

    #@autocast()
    def forward(self, x: torch.Tensor) -> torch.Tensor:
        return self.model(x)

def freeze(
    model,
    freeze_emb=False,
    freeze_ln=False,
    freeze_attn=True,
    freeze_ff=True,
    freeze_other=False,
):
```

```
    for name, p in model.named_parameters():
    # freeze all parameters except the layernorm and positional embeddings
        name = name.lower()
        if 'ln' in name or 'norm' in name:
            p.requires_grad = not freeze_ln
        elif 'embeddings' in name:
            p.requires_grad = not freeze_emb
        elif 'mlp' in name:
            p.requires_grad = not freeze_ff
        elif 'attn' in name:
            p.requires_grad = not freeze_attn
        else:
            p.requires_grad = not freeze_other

    return model

class ClipCaptionModel(nn.Module):
    def __init__(self, prefix_length: int, prefix_size: int = 768):
          super(ClipCaptionModel, self).__init__()
          self.prefix_length = prefix_length
          """
          ru gpts fix, help sometimes

          """
          self.gpt =
          ↪   GPT2LMHeadModel.from_pretrained('sberbank-ai/rugpt3small_based_on_gpt2')

          self.gpt_embedding_size = self.gpt.transformer.wte.weight.shape[1]
          self.clip_project = MLP((prefix_size, (self.gpt_embedding_size *
          ↪   prefix_length) // 2,
                                  self.gpt_embedding_size * prefix_length))

    def get_dummy_token(self, batch_size: int, device: torch.device) -> torch.Tensor:
        return torch.zeros(batch_size, self.prefix_length, dtype=torch.int64,
        ↪   device=device)

    #@autocast()
    def forward(self, tokens: torch.Tensor, prefix: torch.Tensor, mask:
    ↪   Optional[torch.Tensor] = None,
                labels: Optional[torch.Tensor] = None):

        embedding_text = self.gpt.transformer.wte(tokens)
        prefix_projections = self.clip_project(prefix).view(-1, self.prefix_length,
        ↪   self.gpt_embedding_size)

        embedding_cat = torch.cat((prefix_projections, embedding_text), dim=1)
        if labels is not None:
            dummy_token = self.get_dummy_token(tokens.shape[0], tokens.device)
            labels = torch.cat((dummy_token, tokens), dim=1)
        out = self.gpt(inputs_embeds=embedding_cat, labels=labels,
        ↪   attention_mask=mask)
        return out

class ClipCaptionPrefix(ClipCaptionModel):

    def parameters(self, recurse: bool = True):
        return self.clip_project.parameters()

    def train(self, mode: bool = True):
        super(ClipCaptionPrefix, self).train(mode)
```

```
        self.gpt.eval()
        return self
```

Инициализируем модель CLIP и токенайзер для текста

```
device = 'cuda:0'
clip_model, preprocess = clip.load("ViT-L/14@336px", device=device, jit=False)
tokenizer = GPT2Tokenizer.from_pretrained('sberbank-ai/rugpt3small_based_on_gpt2')
```

Инициализируем код предобработки данных, обработанные данные сохраняться в `.pkl`, что позволит не выполнять повторно дорогостоящую операцию преобразования.

```
import io
import os
import PIL
import random
import numpy as np
import torch
import torchvision
import transformers
import more_itertools
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm
import pandas as pd
from torch.utils.data import Dataset
from tqdm import tqdm
from dataclasses import dataclass, field
import torchvision.transforms as T
import torchvision.transforms.functional as TF
import cv2
from PIL import Image
import pickle
from tqdm.contrib import tzip
def image_grid(imgs, rows, cols):
    pils = imgs

    assert len(imgs) == rows*cols

    w, h = imgs[0].size
    grid = Image.new('RGB', size=(cols*w, rows*h))
    grid_w, grid_h = grid.size

    for i, img in enumerate(imgs):
        grid.paste(img, box=(i%cols*w, i//cols*h))
    return grid

def read_video(path, transform=None, frames_num=16, window=30):
    frames = []
    cap = cv2.VideoCapture(path)

    fps = int(cap.get(cv2.CAP_PROP_FPS))

    length = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    N = length//(frames_num)

    current_frame = 0
```

```
    for i in range(length):
        ret, frame = cap.read(current_frame)

        if ret and i==current_frame and len(frames)<frames_num:
            size = 196, 196
            frame = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
            frame.thumbnail(size, Image.ANTIALIAS)

            frames.append(frame)
            current_frame += N

    cap.release()
    return frames

out_path = f"Features_train.pkl"
video_path =  'videos'
import pandas as pd
df = pd.read_csv('ru_train_full.csv')

all_embeddings = []
all_captions = []
i = 0
for video_name, question, answer in tzip(df.video_name, df.question, df.answer):
    name = f'{video_path}/{video_name}.mp4'
    text = f'Вопрос: {question} ? Ответ: {answer}.'
    # обратите внимание, что с этой частью стоит поэкспрементировать
    if os.path.exists(name):
        video = read_video(path = name, frames_num=4)
        if len(video)>1:
            image = image_grid(video,2,2)# можно попробовать меньше/больше кадров

            image = preprocess(image).unsqueeze(0).to(device)
            with torch.no_grad():
                prefix = clip_model.encode_image(image).cpu()
            all_embeddings.append(prefix)
            all_captions.append(text)

with open(out_path, 'wb') as f:
    pickle.dump({"clip_embedding": torch.cat(all_embeddings, dim=0), "captions":
    ↪   all_captions}, f)

print('Done')
print("%0d embeddings saved " % len(all_embeddings))
```

Опишем класс датасета, он наследуется из `torch.dataset`.

```
class ClipCocoDataset(Dataset):
    def __init__(self, data_path: str,  prefix_length= 50, gpt2_type =
    ↪   'sberbank-ai/rugpt3small_based_on_gpt2',
                 normalize_prefix=False):
        self.tokenizer = GPT2Tokenizer.from_pretrained(gpt2_type)
        self.prefix_length = prefix_length
        self.normalize_prefix = normalize_prefix
        with open(data_path, 'rb') as f:
            all_data = pickle.load(f)
        print("Data size is %0d" % len(all_data["clip_embedding"]))
        sys.stdout.flush()
        self.prefixes = all_data["clip_embedding"]
        captions_raw = all_data["captions"]
```

```
        self.captions = captions_raw

        self.captions_tokens = []
        self.caption2embedding = []
        max_seq_len = 0
        i=0
        for caption in tqdm(captions_raw):
            self.captions_tokens.append(torch.tensor(self.tokenizer.encode(caption),
            ↪   dtype=torch.int64))
            self.caption2embedding.append(self.prefixes[i])
            i+=1
            max_seq_len = max(max_seq_len, self.captions_tokens[-1].shape[0])

        all_len = torch.tensor([len(self.captions_tokens[i]) for i in
        ↪   range(len(self))]).float()
        self.max_seq_len = min(int(all_len.mean() + all_len.std() * 10),
        ↪   int(all_len.max()))

    def pad_tokens(self, item: int):
        tokens = self.captions_tokens[item]
        padding = self.max_seq_len - tokens.shape[0]
        if padding > 0:
            tokens = torch.cat((tokens, torch.zeros(padding, dtype=torch.int64) - 1))
            self.captions_tokens[item] = tokens
        elif padding < 0:
            tokens = tokens[:self.max_seq_len]
            self.captions_tokens[item] = tokens
        mask = tokens.ge(0)  # mask is zero where we out of sequence
        tokens[~mask] = 0
        mask = mask.float()
        mask = torch.cat((torch.ones(self.prefix_length), mask), dim=0)  # adding
        ↪   prefix mask
        return tokens, mask

    def __len__(self) -> int:
        return len(self.captions_tokens)

    def __getitem__(self, item):
        tokens, mask = self.pad_tokens(item)
        prefix = self.prefixes[item]
        if self.normalize_prefix:
            prefix = prefix.float()
            prefix = prefix / prefix.norm(2, -1)
        return tokens, mask, prefix
```

Опишем функцию обучения

```
import torch
import torch.nn as nn
from torch.nn import functional as nnf
from torch.utils.data import Dataset, DataLoader
from transformers import GPT2Tokenizer, GPT2LMHeadModel, AdamW,
↪   get_linear_schedule_with_warmup
from tqdm import tqdm
import os
import pickle
import sys
import argparse
import json
```

```python
from typing import Tuple, Optional, Union
from torch.cuda.amp import autocast

# Импортируем необходимые библиотеки
import wandb
wandb.login()
wandb.init(project="clip_caption_video", entity="alexwortega")

# Начинаем определять функцию обучения модели
def train(dataset: ClipCocoDataset, model: ClipCaptionModel, args,
          warmup_steps: int = 2000, output_dir: str = ".", output_prefix: str = ""):

    device =torch.device('cuda:0') # Определяем устройство, на котором будем обучать
    ↪ модель

    batch_size = args.bs # Задаем размер пакета
    epochs = args.epochs # Задаем количество эпох

    if not os.path.exists(output_dir):
        os.makedirs(output_dir) # Создаем папку для сохранения модели

    model = freeze(model) # Замораживаем слои модели
    model = model.to(device) # Переносим модель на устройство

    model.train() # Определяем, что модель находится в режиме обучения
    optimizer = AdamW(model.parameters(), lr=args.lr) # Используем оптимизатор AdamW
    ↪ для обучения модели

    train_dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=True,
    ↪ drop_last=True) # Загружаем данные в DataLoader

    scheduler = get_linear_schedule_with_warmup(
        optimizer, num_warmup_steps=warmup_steps, num_training_steps=epochs *
        ↪ len(train_dataloader)
    ) # Создаем расписание скорости обучения с прогревом

    for epoch in range(epochs):
        print(f">>> Training epoch {epoch}")
        sys.stdout.flush()
        progress = tqdm(total=len(train_dataloader), desc=output_prefix) # Отображаем
        ↪ полосу прогресса
        step=0
        for idx, (tokens, mask, prefix) in enumerate(train_dataloader):
            model.zero_grad() # Обнуляем градиенты
            step+=1

            tokens, mask, prefix = tokens.to(device), mask.to(device),
            ↪ prefix.to(device, dtype=torch.float32) # Переносим данные на
            ↪ устройство

            outputs = model(tokens, prefix, mask) # Получаем выход модели

            logits = outputs.logits[:, dataset.prefix_length - 1: -1] # Отбрасываем
            ↪ лишние данные

            loss = nnf.cross_entropy(logits.reshape(-1, logits.shape[-1]),
            ↪ tokens.flatten(), ignore_index=0) # Вычисляем функцию потерь

            loss.backward() # Распространяем градиенты обратно по модели
            optimizer.step() # Обновляем параметры модели
```

```
            scheduler.step()
            optimizer.zero_grad() # Обнуляем градиенты оптимизатора
            progress.set_postfix({"loss": loss.item()}) # Отображаем значение функции
            ↪   потерь в полосе прогресса

            wandb.log({"loss":  loss.item()}) # Логируем значение функции потерь в
            ↪   WandB
            progress.update() # Обновляем полосу прогресса

            if (idx + 1) % 7000 == 0:
                torch.save(
                    model.state_dict(),
                    os.path.join(output_dir, f"{output_prefix}_latest.pt"),
                ) # Сохраняем модель

        progress.close() # Закрываем полосу прогресса
        if epoch % args.save_every ==0:
            torch.save(
                model.state_dict(),
                os.path.join(output_dir, f"{output_prefix}-{epoch:03d}.pt"),
            ) # Сохраняем модель на каждой эпохе, заданной в аргументах

    return model # Возвращаем обученную модель
# Определяем класс для хранения аргументов командной строки
class Args():
    def __init__(self):
        self.backbone = 'sberbank-ai/rugpt3small_based_on_gpt2'
        self.data = 'Features_train.pkl'
        self.out_dir = 'checkpoints'
        self.prefix = 'prefix_small'
        self.epochs = 2
        self.save_every = 1
        self.prefix_length = 50
        self.bs = 1
        self.only_prefix = False
        self.lr = 2e-5

# Определяем функцию main для запуска обучения модели
def main():
    args = Args() # Создаем объект класса Args и задаем значения аргументов

    wandb.config = {
      "learning_rate": args.lr,
      "epochs": args.epochs,
      "batch_size": args.bs
    } # Логируем аргументы в WandB

    prefix_length = args.prefix_length # Задаем длину префикса

    dataset = ClipCocoDataset(args.data, prefix_length) # Создаем датасет

    model = ClipCaptionModel(prefix_length) # Создаем модель

    print("Train both prefix and GPT")
    sys.stdout.flush()

    train(dataset, model, args, output_dir=args.out_dir, output_prefix=args.prefix) #
    ↪   Обучаем модель
```

Материалы для подготовки

В ходе 1-го этапа для участников был проведен онлайн-интенсив, который позволял в короткие сроки погрузиться в тему искусственного интеллекта и развить необходимые базовые навыки для участия в соревнованиях. Отдельно выделялись темы, связанные с компьютерным зрением и обработкой естественного языка. Материалы интенсива были доступны участникам на протяжении всего соревнования по ссылке: https://stepik.org/course/124175.

Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи инженерного тура и предметного тура по двум предметам: информатике и математике. В каждом предмете максимально можно было набрать 100 баллов, в инженерном туре — 100 баллов. Чтобы пройти во второй этап участники должны были набрать в сумме по обоим предметам предметного тура и инженерном туре не менее 50 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 36 баллов независимо от уровня.

Заключительный этап

В заключительном этапе участники решали задачи предметного и инженерного туров.

Индивидуальный предметный тур

- Информатика — максимально возможный балл за все задачи — 100 баллов.
- Математика — максимально возможный балл за все задачи — 100 баллов.

Командный инженерный тур

Команды заключительного этапа получали за командный инженерный тур от 0 до 100 баллов. Команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем.

Результаты команд по инженерному туру нормировались по формуле:

$$Total = \frac{Result \times (RankParticipants + 1 - RankPlace) \times 100}{MaxResult \times Participants},$$

где $Total$ — итоговый балл за решение задачи;

$Result$ — результат решения задачи;

$RankParticipants$ — общее количество участников в рейтинге;

$RankPlace$ — место участника в рейтинге;

$MaxResult$ — максимальный результат решения задачи среди всех участников;

$Participants$ — общее количество участников.

Формула для построения индивидуального рейтинга участников

В заключительном этапе олимпиады баллы участника складываются из двух частей, каждая из которых имеет собственный вес: баллы за индивидуальное решение задач предметного тура (информатика, математика) с весом $K_1 = 0,1$ каждый предмет и баллы за командное решение задач инженерного тура с весом $K_2 = 0,8$.

Итоговый индивидуальный балл участника определяется по формуле:

$$S = K_1 \cdot (S_1 + S_2) + K_2 \cdot S_3,$$

где S_1 — балл предметного тура заключительного этапа по математике в стобалльной системе ($S_{1_{\text{макс}}} = 100$);

S_2 — балл предметного тура заключительного этапа по информатике в стобалльной системе ($S_{2_{\text{макс}}} = 100$);

S_3 — итоговый нормированный балл инженерного командного тура в стобалльной системе ($S_{3_{\text{макс}}} = 100$).

Итого максимально возможный индивидуальный балл участника заключительного этапа — 100 баллов.

Критерий определения победителей и призеров

Чтобы определить победителей и призеров независимо от уровня (класса) на основе индивидуальных результатов участников был сформирован общий рейтинг всех участников заключительного этапа. Сначала рейтинга были выбраны 8 победителей и 17 призёров (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	53 и выше
Призеры	От 36 до 52

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.

Важная задача наставника — превратить неудачу в инструмент будущего успеха. Для этого необходимо вместе с учениками наметить план развития компетенций и подготовки к будущему сезону Олимпиады. 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). Такие опросники 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.

Еще одним направлением работы наставника после НТО может стать создание кружка по направлению профилей или по формированию необходимых компетенций: программирование, электроника, робототехника, 3D-моделирование и т. п. Формат подобного кружка может быть различным: короткие модули, дополнительные курсы, факультативы, группы дополнительного образования. Для создания кружков можно воспользоваться образовательными программами, опубликованными на сайте НТО: https://ntcontest.ru/mentors/education-programs/.

Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с родителями помогает наставнику продемонстрировать им 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».