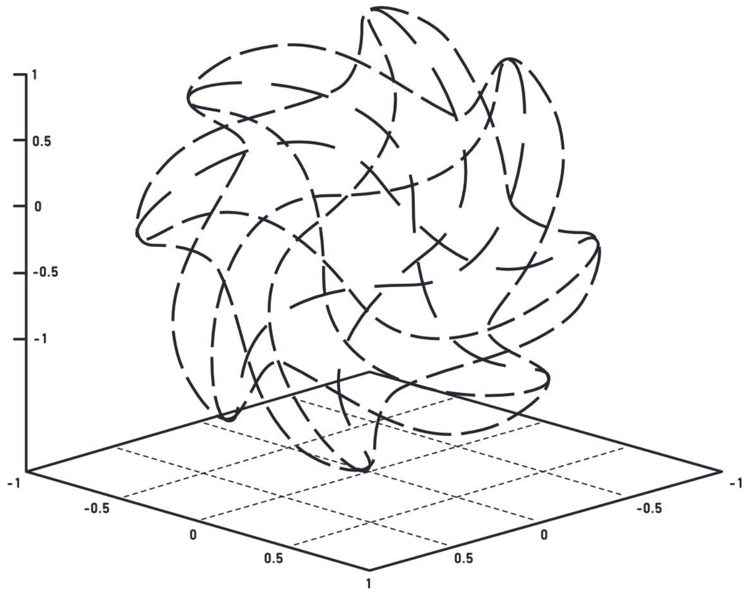


МАТЕРИАЛЫ ЗАДАНИЙ
командной инженерной олимпиады школьников «Олимпиада Кружкового движения Национальной технологической инициативы»

по профилю «Искусственный интеллект»

2019/20 учебный год

http://nti-contest.ru

Оглавление

Введение

Олимпиада Кружкового движения Национальной технологической инициативы (далее – Олимпиада КД НТИ) – это командная инженерная олимпиада школьников, завершающаяся разработкой действующего устройства, системы устройств или компьютерной программы. Олимпиада является проектом Агентства стратегических инициатив, элементом дорожной карты НТИ «Кружковое движение» и ключевым механизмом вовлечения школьников в образовательные программы высшего образования, ориентированные на рынки НТИ. Оператором Олимпиады КД НТИ является некоммерческая организация «Ассоциация участников технологических кружков».

Олимпиада КД НТИ является одним из этапов работы по реализации дорожной карты НТИ «Кружковое движение». Подготовка дорожной карты велась в фаблабах, ЦМИТах, детских технопарках, на базе активных школ и лицеев, центров дополнительного образования по всей России. Цели рабочей группы «Кружковое движение» НТИ связаны с развитием технологического сообщества, объединяющего школьников и студентов, ориентированных на инженерную деятельность на рынках НТИ, самодеятельных технических энтузиастов, лидеров технологических кружков, разработчиков педагогических технологий, технологических предпринимателей, популяризаторов науки и технологий. На сайте «Кружкового движения» НТИ собраны категоризированные подборки материалов для учащихся разных классов и запущен проект wiki-библиотеки образовательных материалов: https://kruzhok.org/.

В настоящее время во исполнение поручения Президента Российской Федерации В.В. Путина Правительством Российской Федерации совместно с Ассоциацией участников технологических кружков прорабатывается вопрос о создании сети кружков на базе общеобразовательных организаций по модели Кружкового движения НТИ: https://regnum.ru/news/economy/2918327.html.

Профили Олимпиады КД НТИ выбраны на основе приоритетов Национальной технологической инициативы: «Аэрокосмические системы», «Автоматизация бизнес-процессов», «Автономные транспортные системы», «Анализ космических снимков и геопространственных данных», «Беспилотные авиационные системы», «Большие данные и машинное обучение», «Водные робототехнические системы», «Интернет вещей», «Инженерные биологические системы: агробиотехнологии и геномное редактирование», «Интеллектуальные робототехнические системы», «Интеллектуальные энергетические системы», «Информационная безопасность», «Искусственный интеллект», «Композитные технологии», «Научно-инженерная коммуникация», «Наносистемы и наноинженерия», «Надводные робототизированные аппараты», «Нейротехнологии и когнитивные науки», «Передовые производственные технологии», «Программная инженерия финансовых технологий», «Разработка приложений виртуальной и дополненной реальности», «Разработка игр», «Спутниковые системы (Системы связи и дистанционного зондирования Земли)», «Технологии беспроводной связи», «Умный город», «Урбанистика».

Целевыми победителями Олимпиады КД НТИ являются школьники, способные реализовать сложные технические проекты в прорывных областях. Олимпиада выделяет команды участников с особыми характеристиками мышления, коммуникации и действия, необходимыми для решения задач НТИ. Победители и призеры Олимпиады КД НТИ должны показать высокие результаты в области применения предметных знаний в практической работе. Одновременно с этим, система подготовки Олимпиады КД НТИ должна предоставлять участникам инструменты для подготовки и получения недостающих знаний и практических навыков.

Цель Олимпиады КД НТИ – поддержка школьников в стремлении решать технологические вызовы XXI века через включение в решение технологических задач переднего края и, одновременно, через повышение социальной значимости такой работы благодаря льготам к поступлению. Эта цель лежит в рамках миссии Кружкового движения: формирование и подготовка команд, способных запускать глобальные технологические проекты, менять мир, создавая новые общественные практики.

Важной особенностью олимпиады является командный формат отборочного и заключительного этапов. Команды формируются на основе компетентностного принципа, различные компетенции участников в одной команде позволяют найти оригинальное нестандартное решение задачи. В командах участники планируют свою работу, обсуждают, ищут решения, распределяют роли – часто один участник выполняет несколько ролей. Комплексные инженерные задачи разработаны таким образом, что их можно декомпозировать на несколько подзадач, за решение которых берутся участники согласно своей роли в команде. Каждый участник несет ответственность за результат работы команды.

Организаторы Олимпиады КД НТИ соблюдают принцип равных возможностей и доступности участия школьников с ограниченными возможностями здоровья. В олимпиаде беспрепятственно могут участвовать дети с ОВЗ, способные выполнять инженерные работы и работать в команде, а также те, кто обучался по состоянию здоровья на дому. Организаторы также заинтересованы в дальнейшем сопровождении ее участников, а школьники – участники Олимпиады КД НТИ заинтересованы в дальнейшем сотрудничестве. В организации заключительного этапа Олимпиады КД НТИ 2019/20 учебного года в качестве волонтеров приняли участие победители и призеры Олимпиады КД НТИ прошлых лет, студенты первых курсов из различных регионов России.

В рамках интеграции выпускников и финалистов в проведение заключительных этапов были опробованы форматы взаимодействия амбассадоров и стажеров Кружкового движения с финалистами Олимпиады КД НТИ, а также выстроена система интеграции участников прошлых лет в сообщество Кружкового движения.

Олимпиада КД НТИ в 2015–2020 гг.

Олимпиада КД НТИ впервые состоялась в 2015/16 учебном году. За период с 2015 по 2020 год количество зарегистрированных участников выросло с нескольких тысяч до 58 тысяч, а количество школьников, приглашенных к участию в заключительном этапе – со 100 до 1262 человек. Количество профилей выросло в пять раз – с четырех до 30. В 2016/17 учебном году четыре профиля Олимпиады КД НТИ впервые вошли в Перечень олимпиад школьников, что позволило победителям и призерам воспользоваться льготами при поступлении в вузы России (в зависимости от правил приема конкретного вуза), в 2017/18 году число включенных в Перечень профилей выросло до девяти, в 2018/19 году – до 13, в 2019/20 учебном году – до 16 (семь – II уровня, девять – III уровня).

В 2017/18 учебном году заключительный этап Олимпиады КД НТИ стал распределенным и проходил в течение нескольких месяцев (с февраля по апрель) на площадках вузов по всей России: ОЦ «Сириус», МАИ, МИФИ, ТПУ, Университет Иннополис, СПбПУ, ДВФУ, УрФУ. В 2018/19 учебном году распределенный финал Олимпиады КД НТИ приняли МФТИ, МАИ, МИФИ, ТПУ, Университет Иннополис, СПбПУ, ДВФУ, НГУ, НовГУ, Московский Политех, ИГУ, ИрНИТУ и ряд других площадок. Также в 2018/19 учебном году впервые были проведены синхронные по времени распределенные финалы на площадках в разных городах в рамках одного профиля. Участники распределенных финалов решали одинаковые задания, имели одинаковые критерии оценивания и единый рейтинг участников.

В 2019/20 учебном году до 17 марта финалы проходили в очном формате в Новосибирске, Иркутске, Владивостоке, Сочи, Москве, Тюмени, Иннополисе, Томске, Санкт-Петербурге. После 17 марта в связи с эпидемиологической обстановкой и необходимостью принятия мер по предупреждению распространения новой коронавирусной инфекции Оргкомитет Олимпиады принял решение о проведении оставшихся заключительных этапов в формате, позволяющем участникам не покидать пределы города проживания и не принимать участия в массовых мероприятиях. Для этого была проведена комплексная оценка и разработаны меры по переводу всех оставшихся профилей в распределенный формат. Предметный тур заключительного этапа и апелляция проводились с применением системы прокторинга. Для организации командного тура использовались разные схемы доступа участников к лабораториям и оборудованию:

- работа с «аватарами»: лаборанты и технические специалисты выполняли роль удаленных операторов и действовали по инструкции от участников, участники видели оборудование через видеокамеры;
- удаленное управление компьютером, к которому подключено оборудование: участники подключались к компьютеру разработчиков и управляли оборудованием, наблюдая за ним с помощью видеокамер;
- разработка 3D-моделей полезной нагрузки и исполняемых программ для управления устройствами запускаемых на компьютерах разработчиков профилей или загружаемых в устройства;
- работа с виртуальными лабораториями и стендами;
- работа с общедоступным оборудованием и расходными материалами, пересылка решений разработчикам.

Всего в новом распределенном формате прошли 14 финалов в период с 17 марта по 31 апреля 2020 года. Для поддержки финалистов, проживающих в разных часовых поясах, и организации командной работы были привлечены модераторы, которые регулярно общались с участниками, контролируя их эмоциональное состояние, стимулируя вовлеченность, и давали обратную связь разработчикам профиля.

Для работы с участниками в новых обстоятельствах организаторы Олимпиады КД НТИ разработали рекомендации по управлению коммуникацией удаленных команд – материал опубликован на сайте Министерства просвещения РФ: `https://edu.gov.ru/press/2324/ministerstvo-prosvescheniya-rekomenduet-ispolzovat-principy-distancionnoy-raboty-komand-predlozhennye-kruzhkovym-dvizheniem-nti/`.

Структура отбора участников Олимпиады КД НТИ

Соревнование проходит в три этапа. Первый отборочный этап проводится в заочной форме для всех зарегистрированных школьников. Для проведения этапа используется интернет-платформа «Stepik» (http://stepik.org) интегрированная с личным кабинетом на сайте олимпиады. Участники решают олимпиадные задания по выбранным предметам (математика, информатика, физика, химия, биология, география, русский язык; таблица соответствия профилей и предметов: https://nti-contest.ru/materials/) согласно уровню: 8-9 классы и 10-11 классы. В зачет идет результат наилучшей из трех доступных попыток. Проверка решений производится на платформе автоматически. По итогам первого отборочного этапа устанавливаются проходные баллы по каждому профилю. Если у участника не хватило баллов на желаемый профиль, он может выбрать другой.

Второй отборочный этап также проводится в заочной форме на интернет-платформе «Stepik» и с использованием инженерных онлайн-симуляторов и виртуальных стендов на платформах технологических компаний или вузов – разработчиков профилей. Участники, прошедшие во второй отборочный этап, решают задачи по тематике профиля. В большинстве профилей на этом этапе требуется работа участников в командах. Этот этап является не только отборочным, но и обучающим мероприятием. Во время проведения второго этапа формируются знания и умения, связанные с выбранным профилем, а также сквозные компетенции: программирование на языках Python и C++, электроника, схемотехника и работа с микроконтроллерами, алгоритмы компьютерного зрения, 3D-моделирования и САПР.

Отборочные этапы сопровождаются подготовительными мероприятиями, разработанными профилями: дистанционными мероприятиями (вебинары); мероприятиями для самостоятельной подготовки (онлайн-курсы); мероприятиями, направленными на получение практических навыков (интенсивы). Для формирования команд создан специальный онлайн-сервис на платформе Олимпиады КД НТИ, а также проводятся мероприятия, направленные на командообразующую деятельность (специальные встречи, интенсивы, очные курсы на площадках по подготовке). Подготовительные мероприятия доступны для всех желающих и разработаны таким образом, чтобы их можно было провести на минимальном количестве оборудования. Часто такие мероприятия проводятся на площадках региональных партнеров со статусом «Методическая площадка» или «Площадка подготовки». Информация о партнерских площадках размещена в специальном разделе официального сайта олимпиады: http://nti-contest.ru/places_to_prepare/.

По итогам второго отборочного этапа список участников заключительного этапа определяется на основании рейтинга и количества мест в финале. Количество мест в финале каждого профиля определяется вместимостью площадки и количеством оборудования для проведения очных соревнований.

Заключительный этап олимпиады состоит из двух частей: индивидуальное решение предметных олимпиадных задач по предметам выбранным профилем и командная разработка инженерного решения с его последующим испытанием. Задание второй части (инженерная задача) заключительного этапа имеет свою специфику для каждого профиля.

Подготовка участников

Для вовлечения участников в олимпиаду были разработаны «Урок НТИ» и «Демо-этап», благодаря чему участники могли определиться с выбором профилей и попробовать свои силы.

«Урок НТИ» (http://nti-contest.ru/ntilessonteacher/) – это инициатива, созданная в сентябре 2018 года и направленная на распространение информации об НТИ среди школьников и привлечение их к Олимпиаде КД НТИ путем проведения уроков и занятий в школах и учреждениях дополнительного образования. Учебный материал для проведения «Урока НТИ» сформирован в виде конструктора, с помощью которого учителя могут собрать урок по теме НТИ. Для участия в проекте «Урок НТИ» зарегистрировалось 3260 педагогов и образовательных организаций.

«Демо-этап» Олимпиады КД НТИ (https://stepik.org/course/24389/) – это опубликованная подборка задач олимпиады. Демо-этап знакомит с профильными задачами, позволяет попробовать решать инженерные задачи. Перед регистрацией и выбором профиля потенциальные участники и их наставники могут познакомиться с задачами и выбрать наиболее интересный для себя профиль.

Сборники задач прошлых лет по всем профилям с 2015 по 2020 годы с решениями задач отборочных и финальных этапов и критериями оценки размещены на сайте Олимпиады КД НТИ: https://nti-contest.ru/problembooks/. Также команды разработчиков профилей подготовили видеоразборы задач 2 этапа, которые размещены в видеоканале олимпиады: https://www.youtube.com/channel/UCZV1CNpOrDNj7tuWuf35lgw/playlists.

На платформе «Stepik» собраны онлайн-курсы по решению олимпиадных заданий для подготовки к Олимпиаде КД НТИ. Эти курсы объединяют практически все задания отборочных этапов и часть заданий финала олимпиады за все время ее существования:

- https://stepik.org/course/1296/ – 2015/16 учебный год;
- https://stepik.org/course/3598/ – 2016/17 учебный год;
- https://stepik.org/course/15697/ – 2017/18 учебный год;
- https://stepik.org/course/55997/ – 2018/19 учебный год;
- https://stepik.org/course/71851/ – 2019/20 учебный год.
- https://stepik.org/course/24389/ – пробный этап для ознакомления с задачами Олимпиады КД НТИ по тематическим кластерам: «Информация», «Природа» «Производство», «Стратегия», «Техника» и «Человек».

Все указанные материалы находятся в свободном доступе и размещены на официальном сайте олимпиады, на страницах профилей, а также в разделе «Материалы для участников» с разбивкой по предметам и профилям https://nti-contest.ru/materials/.

Организаторы и партнеры Олимпиады КД НТИ

Оргкомитет Олимпиады представлен ректорами крупнейших политехнических и инженерных вузов России, руководителями технологических компаний и представителями государственных органов.

Вузы-соучредители олимпиады:

- ФГБОУ ВО «Московский политехнический университет»;
- ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого»;
- ФГАОУ ВО «Национальный исследовательский Томский политехнический университет»;
- ФГАОУ ВО «Дальневосточный федеральный университет»;
- АНО ВО «Университет Иннополис».

Технологические партнеры

Олимпиада КД НТИ проводится при поддержке партнеров – технологических компаний, присутствующих на рынке инновационных технологий: ПАО «Аэрофлот», Благотворительный фонд Сбербанка «Вклад в будущее», ПАО «Ростелеком», фирма «1С», ФИОП РОСНАНО, ПАО «Газпром нефть», компания «Movavi», ПАО «ОАК», ПАО «Компания Сухой», госкорпорация «Роскосмос», компания «Bayer», Фонд новых форм развития образования, сеть детских технопарков «Кванториум», компания «Спутникс», компания «Полюс-НТ», компания «BiTronicsLab», компания «КРОК», компания «Инфосистемы Джет», компания «Лоретт», компания «COEX», Академия Высоких Технологий, компания «Образование будущего» и др. Полный список организаторов и партнеров олимпиады размещен в соответствующем разделе на официальном сайте: http://nti-contest.ru/about/.

Популяризация Олимпиады КД НТИ

В период с 15 августа 2019 г. по 01 апреля 2020 г. об Олимпиаде КД НТИ вышло 5174 публикации. 1571 публикация вышла в СМИ федерального уровня, 3548 – в СМИ на региональном уровне, 51 – в зарубежных СМИ. Охват из открытых источников: 450,4 млн.; МедиаИндекс – 27,805; количество реакций на публикации олимпиады в социальных сетях (likes и shares) – 9234. Об Олимпиаде КД НТИ вышло более 10 телевизионных сюжетов на федеральных и региональных каналах: «Россия - 1», «Россия – Культура», «Телеинформ», ГТРК «Амур», «Новгородское областное телевидение», «Аист-ТВ», «Телеканал ТюмГУ», «Удмуртия-24» и др.

Во время проведения отборочных этапов Олимпиада КД НТИ освещалась в федеральных, массовых, родительских, образовательных и других медиа («ИТАР-ТАСС», «РИА-Новости», «Интерфакс», «Такие Дела», Letidor, «Дети Mail.ru», «Индикатор», «Занимательная робототехника», Rusbase, «Учёба.Ру»), на официальных образовательных порталах и порталах органов государственности власти в регионах. На радио «MediaMetrics» регулярно выходила программа «Выше среднего», гостями выступали разработчики профилей олимпиады и ее партнеры. Кампания по привлечению шла также в научно-популярных группах и группах вузов и площадок партнеров.

Заключительный этап Олимпиады КД НТИ в 2019/20 учебном году проходил при поддержке федеральных и региональных СМИ, телевизионных каналов, популярных молодежных и социальных порталов. В период проведения финалов на очных площадках большинство представителей профилей подготовили ролики и фотографии, которые активно распространялись в социальных сетях.

Широкое освещение мероприятий заключительного этапа имеет целью распространение информации среди потенциальных участников Олимпиады КД НТИ будущего года и привлечение талантливых школьников со всей России. Список лучших материалов об олимпиаде: http://nti-contest.ru/publications/.s

Профиль «Искусственный интеллект»

Профиль Олимпиады Кружкового движения НТИ 2019/20 учебного года «Искусственный интеллект» знакомил школьников с методами машинного обучения, используемыми для предсказания пользовательского поведения. Данная задача является одной из ключевых для исследователей данных и находит применение во многих сферах деятельности человека: энергетике, транспорте, госуправлении, логистике, торговле, услугах, развлечениях и, конечно, банках. Учащиеся 8–11 классов, прошедшие все этапы олимпиады, демонстрировали 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.

Первый отборочный дистанционный этап (индивидуальный) определял общий уровень подготовки школьников по предметам математика и информатика. Решая задачи по информатике из раздела программирование, школьники должны были продемонстрировать простейшие навыки составления и отладки программ, обрабатывающих массивы данных, и понимание таких тем, как комбинаторика, операции со строками, вычислительная геометрия, теория графов. Задачи по математике проверяли у участников знания по алгебре, теории вероятностей, геометрии. Количество попыток сдачи решения задач не ограничивалось. Таким образом, задачи первого этапа выявляли наличие у участников знаний необходимых не только для решения задач следующего этапа, но и финальной задачи.

Задача **второго отборочного дистанционного этапа** заключалась в формировании навыков обработки данных и выявления признаков, необходимых для решения финальной задачи, а также в отборе наиболее сильных участников, поэтому задача решалась каждым участником самостоятельно. Для поддержки и развития участников был проведен вебинар с разработчиком задачи, где было подробно разобрано базовое решение задачи, а также участники могли задать свои вопросы и получить ответы. Помимо этого, участникам были предоставлены материалы для формирования навыков в области программирования на Python и использования основных библиотек для анализа данных, основам машинного обучения, теории вероятностей, практикумы и сборники задач с предыдущих соревнований.

Эти знания позволили участникам справиться с задачей, где требовалась обработка больших объемов данных (около 27 000 000 записей), и разработать модель машинного обучения, решающую задачу классификации.

Командная задача **очного заключительного этапа** стала следующим логичным шагом в работе с выбранной темой. Участникам предстояло предсказать вероятность совершения покупки в каждой из 8 выбранных категорий клиентами Сбербанка в течение ближайших 7 дней. Для решения данной задачи было предложено 2 набора обезличенных данных о транзакциях 50 000 клиентов банка в течение одного года. Один набор предназначался для обучения модели, второй – для проверки. Каждый из них содержал данные о транзакциях 25 000 различных клиентов.

Также на заключительном этапе участники приняли участие индивидуальном предметном туре, в ходе которого участники решали задачи по математике и информатике. Задачи по математике покрывали следующие области математики: оптимизация, комбинаторика, алгебра и геометрия. А темы задач по информатике перекликались с классическими темами всероссийской олимпиады школьников.

Первый отборочный этап

Задачи первого этапа. Информатика

Первая попытка

Задача I.1.1.1. Фанатам стратегий (20 баллов)

Миссия простая: нужно либо накопить c кредитов (читайте: единиц вымышленной валюты), либо сокрушить врага в этом регионе. Миссия проходная и не интересная (даже никаких юнитов кроме пехоты), так что надо её закрыть побыстрее.

Изначально у нас нет ни одного кредита и ни одного отряда пехоты. Но есть база и гарантия того, что враг нас не обнаружит, пока мы на него не нападём.

На нашей базе можно хранить не более x кредитов. Чтобы хранить больше, нужно строить хранилища: каждое стоит y ($y \leqslant x$) кредитов и в каждом можно будет хранить дополнительно по максимум z кредитов. Строительство моментально (а вы что, реализма ожидали?).

Через каждые m минут нам привозят добычу на s кредитов. Добыча моментально конвертируется в кредиты и мы перераспределяем новые s кредитов и кредиты, имеющиеся на базе, по трём направлениям: строительство хранилищ, наём отрядов пехоты, хранение на базе. Количество кредитов, отданных на строительство хранилищ, должно быть кратно стоимости постройки одного хранилища. Количество кредитов, отданных на наём отрядов пехоты, должно быть кратно стоимости наёма одного отряда пехоты. Если мы отдаём на хранение больше кредитов, чем может храниться на базе (хранилища, на строительство которых мы только что отдали кредиты, тоже учитываются), то излишки кредитов исчезают.

Если после очередного описанного выше распределения на базе будет храниться хотя бы c кредитов, то считается, что мы накопили c кредитов, и миссия считается пройденной.

Один отряд пехоты стоит f ($f \leqslant x$) кредитов. Сразу после оплаты, которая тоже моментальна, отряд будет ждать приказаний на нашей базе.

Проще и быстрее всего избавиться от врага в этом регионе – заставить его капитулировать. Для этого нужно привести к стенам вражеской базы больше отрядов пехоты, чем есть на вражеской базе.

По данным разведки, на вражеской базе e отрядов, а добраться до вражеской базы наши отряды пехоты смогут за t минут (все отряды, независимо от их количества, могут идти вместе).

За сколько в лучшем случае мы завершим миссию?

Формат входных данных

Единственная строка содержит 9 целых чисел $c, x, y, z, m, s, f, e, t$ ($1 \leqslant c, x, y, z, m, s, f, e, t \leqslant 10^9$; $f, y \leqslant x$).

Формат выходных данных

Выведите одно целое число – искомое количество минут.

Пояснения

Чтобы максимально быстро накопить 2700 кредитов, мы можем построить 2 хранилища через 3 минуты после начала (в этот момент у нас появятся первые кредиты) и далее просто складировать все кредиты на базе. Тогда через 9 минут после начала на базе накопится необходимая сумма.

Чтобы заставить врага сдаться, мы можем через 3 минуты после начала нанять 10 отрядов пехоты и сразу же пойти на вражескую базу. Тогда через 6 минут после начала враги сдадутся.

Примеры

Пример №1

Стандартный ввод
`2700 1000 150 1000 3 1000 100 7 3`

Стандартный вывод
`6`

Решение

В данной задаче нужно выбрать быстрейший путь: накопить c кредитов или сокрушить врага.

Рассмотрим накопление. В этом случае для минимизации времени стоит тратить кредиты только на постройку хранилищ. Так как стоимость одного хранилища не превышает изначального максимального количества кредитов, которое мы можем хранить на базе, то есть $y \leqslant x$, при поставке добычи мы всегда можем распределить кредиты так, что мы отдаём на хранение меньше кредитов, чем может храниться на базе.

Таким образом, если нужны хранилища, то есть $c > x$, достаточно построить $\left\lceil \frac{c-x}{z} \right\rceil$ хранилищ. Получается, нужно набрать в сумме $c + \left\lceil \frac{\max(0, c-x)}{z} \right\rceil \cdot y$ кредитов, что мы сделаем за $\left\lceil \frac{c + \left\lceil \frac{\max(0, c-x)}{z} \right\rceil \cdot y}{s} \right\rceil \cdot m$ минут.

Рассмотрим капитуляцию врага. В этом случае для минимизации времени стоит тратить кредиты только на наём отрядов пехоты. Так как стоимость одного отряда пехоты не превышает изначального максимального количества кредитов, которое мы можем хранить на базе, то есть $f \leqslant x$, при поставке добычи мы всегда можем распределить кредиты так, что мы отдаём на хранение меньше кредитов, чем может храниться на базе.

Чтобы заставить врага капитулировать, достаточно нанять $e+1$ отрядов пехоты и привести все эти отряды к стенам вражеской базы. Для этого придётся потратить в сумме $(e+1)\cdot f$ кредитов на пехоту, что мы сделаем за $\left\lceil \frac{(e+1)\cdot f}{s} \right\rceil \cdot m$ минут. Плюс t минут на то, чтобы добраться до врага. Итого $\left\lceil \frac{(e+1)\cdot f}{s} \right\rceil \cdot m + t$ минут.

Минимум из двух описанных значений и является ответом. Также стоит отметить, что 64-битных типов данных для полного решения будет недостаточно.

Пример программы-решения

Ниже представлено решение на языке Python3

```
c, x, y, z, m, s, f, e, t = [int(i) for i in input().split()]
print(min((c + (max(0, c - x) + z - 1) // z * y + s - 1) // s * m,
          ((e + 1) * f + s - 1) // s * m + t))
```

Задача I.1.1.2. Настроение профессора (20 баллов)

Все персонажи и описываемые события являются вымышленными. Любое совпадение с реальными людьми или событиями случайно.

Завтра студенты первого курса пойдут сдавать математический анализ. Экзамен будет принимать профессор Ильдар.

Экзамен будет проходить по старинке: студенты по одному подходят к профессору, отвечают на заданные им вопросы и получают свои оценки. Результат экзамена сильно зависит от настроения профессора Ильдара: если у него плохое настроение, то не важно, насколько хорошо вы подготовились, – он отправит вас на пересдачу.

Пусть настроение профессора в некоторый момент времени равно x. После ответов отличника настроение профессора повышается и становится равно $x+1$. После ответов хорошиста настроение профессора не меняется. А если ответы явно не тянут на оценку 4, то профессор ставит 3 и его настроение падает до $x-1$.

Но если завтра в какой-либо момент времени настроение профессора будет равно отрицательному числу, то после этого момента описанные выше закономерности перестают действовать и все студенты, что ещё не получили своих оценок, отправляются на пересдачу.

Сегодня вы (неожиданно) – староста группы и хотите, чтобы никто из ваших студентов не отправился на пересдачу. Порядок, в котором студенты будут подходить к профессору, уже сформирован и его изменить нельзя, но вы знаете, насколько хорошо подготовился каждый из студентов, и знаете про профессора Ильдара ещё одну вещь – он любит шоколад.

Вы можете купить шоколадку (а лучше не одну) и подарить её профессору сегодня вечером. Каждая подаренная профессору шоколадка повышает его настроение на 1. Что профессор делает с шоколадками, никому не известно.

Какое минимальное количество шоколадок вам надо сегодня подарить профессору, чтобы завтра все студенты сдали экзамен?

Формат входных данных

В первой строке вводятся два целых числа n и k $(1 \leqslant n \leqslant 2 \cdot 10^5, -10^9 \leqslant k \leqslant 10^9)$ – количество студентов в вашей группе и настроение профессора сегодня вечером (настроение профессора до начала экзамена может измениться только благодаря вам).

Во второй строке вводится строка из n символов a_i $(a_i \in \{A, B, C\})$. Эта строка описывает порядок, в котором студенты будут подходить к профессору. Каждый студент описывается одним символом. Символом A обозначается отличник, символом B – хорошист, символом C – троечник или неподготовившийся к экзамену студент.

Формат выходных данных

Выведите одно целое число – искомое минимальное количество шоколадок.

Примеры

Пример №1

Стандартный ввод
`3 0` `BCA`

Стандартный вывод
`1`

Пример №2

Стандартный ввод
`3 3` `AAA`

Стандартный вывод
`0`

Решение

Промоделируем сдачу экзамена без учёта отправки всех на пересдачу. Найдём минимальное настроение профессора в день экзамена (учитывая настроение до начала и не учитывая после окончания). Если оно отрицательно и равно x, то достаточно купить профессору $-x$ шоколадок перед экзаменом, чтобы никто не был отправлен на пересдачу. Если оно неотрицательно, то можно обойтись без подарков профессору.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
n, k = [int(i) for i in input().split()]
min_k = 10 ** 9 + 1
for c in input():
    min_k = min(k, min_k)
    k += ord('B') - ord(c)

print(max(0, -min_k))
```

Задача I.1.1.3. Комната ярости (20 баллов)

Гертруда имеет n тарелок. И хочет разбить их все. По одной. Но тарелки бьются очень звонко. Она опасается, что повредит слух.

Известно, что сила звона первой разбитой тарелки будет равна a_1. Сила звона каждой последующей разбитой тарелки будет в b раз больше силы звона предыдущей. То есть сила звона i-ой ($i > 1$) разбитой тарелки будет равна $a_i = a_{i-1} \cdot b$.

Гертруда знает максимальное суммарное значение сил звона MAX, которое могут выдержать её уши, и желает максимально насладиться звуками бьющейся посуды.

Помогите Гертруде, найдите максимальное количество тарелок, которые она может разбить, не повредив слух. И побыстрее.

Формат входных данных

В первой строке вводятся четыре целых числа n, a_1, b, MAX ($1 \leqslant n, MAX \leqslant 10^{1000}, 1 \leqslant a_1, b \leqslant 10$).

Формат выходных данных

Выведите максимальное количество тарелок.

Пояснения

Если Гертруда разобьёт одну тарелку, то суммарное значение сил звона будет равно $a_1 = 1$.

Если разобьёт две тарелки, то суммарное значение будет равно $a_1 + a_2 = 1 + 2 = 3$.

Если разобьёт три, то – $a_1 + a_2 + a_3 = 1 + 2 + 4 = 7$.

Примеры

Пример №1

Стандартный ввод
`10 1 2 4`

Стандартный вывод
`2`

Решение

В данной задаче для нахождения ответа можно было воспользоваться формулой суммы первых k элементов геометрической прогрессии и бинпоиском по ответу, но ограничения на входные данные устроены так, что можно было просто промоделировать разбивание тарелок по одной.

Для работы с длинными целыми числами стоило либо использовать соответствующий встроенный тип данных, либо писать свою реализацию длинной арифметики.

Первые подзадачи решались без длинной арифметики.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n, ai, b, max = [int(i) for i in input().split()]

if b == 1:
    print(min(max // ai, n))
    exit(0)

# b > 1
l, s = 1, ai
while l <= n:
    if s > max:
        print(l - 1)
        exit(0)
    ai *= b
    s += ai
    l += 1

print(n)
```

Задача I.1.1.4. Подлизы (20 баллов)

Жила-была девочка Катя, и было у неё много-много денег. И подруг. Ну как подруг...

И собрались они как-то раз все вместе у Кати дома и обсуждали фильмы. Многие хвалили вкус Кати. Редко кто не соглашался с её мнением. О вкусах, конечно, не спорят, но Кате показалось это странным и она решила устроить проверку.

Катя записала m пар фильмов, которые девочки сравнивали, и для каждой такой пары пометила, какой из фильмов девочки посчитали однозначно лучше другого. А потом воспользовалась своим обаянием влиянием и убедила школьного психолога провести тестирование, в котором есть вопрос о трёх любимых фильмах. Вот так вот всё просто, когда ты – Катя.

Среди неиспорченных бланков тестирования (не спрашивайте, как она их достала) Катя нашла заполненные бланки n своих подруг. Скажите, согласовываются ли записи Кати с каждым из ответов на вопрос о трёх любимых фильмах в отдельности.

Формат входных данных

В первой строке заданы числа n и m ($1 \leqslant n, m \leqslant 10^5$).

В следующих m строках – пары фильмов, записанные у Кати. Первый фильм в паре считается лучше второго.

В следующих n строках – списки любимых фильмов девочек. Первый фильм в тройке считается лучше второго, а второй – лучше третьего.

Записи Кати непротиворечивы. Каждая пара фильмов в записях Кати встречается не более одного раза.

Так сложилось, что все фильмы, что встречаются в списках любимых фильмов девочек, встречаются и в записях Кати, а в каждом отдельно взятом списке все три фильма различны.

Для вашего же удобства названия фильмов во входных данных заменены на положительные натуральные числа, не превышающие 10^6.

Формат выходных данных

Выведите n строк, в i-ой из которых должно быть написано $honest$, если список любимых фильмов из i-го бланка не противоречит записям Кати, или $liar$, если противоречит.

Не выводите лишние пробелы в конце или начале строк - это будет считаться за ошибку.

Пояснения

Тройка фильмов 1 2 4 противоречит записям Кати, так как по записям Кати фильм 5 лучше фильма 4, но его нет в тройке.

Тройка фильмов 1 3 2 противоречит, так как по записям Кати фильм 2 лучше фильма 3, а в тройке фильм 3 стоит до фильма 2.

Тройка фильмов 5 4 8 противоречит, так как по записям Кати фильм 2 лучше фильма 4, но его нет в тройке.

Примеры

Пример №1

Стандартный ввод

```
5 8
1 3
1 2
2 3
2 4
4 8
5 4
5 6
7 6
1 2 3
1 2 4
1 3 2
5 4 8
5 7 6
```

Стандартный вывод

```
honest
liar
liar
liar
honest
```

Решение

Из условия следует, что тройка любимых фильмов подруги Кати может противоречить записям Кати одним из двух способов (или сразу обоими способами):

1. Существует фильм, который лучше одного из тройки, но в эту тройку не включён;
2. 2 фильма из тройки сравнивали и фильм, который хуже по записям Кати, оказался в тройке на месте выше, чем фильм, который лучше по записям Кати.

Также можно вывести определение того, какая тройка фильмов не противоречит записям Кати.

Такая тройка удовлетворяем всем следующим условиям:

1. Не существует фильма, который по записям Кати лучше первого фильма в тройке;
2. Фильмов, которые по записям Кати лучше второго фильма в тройке,
либо не существует,
либо такой фильм только один и это первый фильм в тройке;
3. Фильмов, которые по записям Кати лучше третьего фильма в тройке,
либо не существует,
либо такой фильм только один и это первый фильм в тройке,
либо такой фильм только один и это второй фильм в тройке,
либо таких фильм ровно два и это первый и второй фильмы в тройке.

Остаётся лишь аккуратно реализовать проверку каждой из троек фильмов.

Асимптотика: $O(m + n)$

Пример программы-решения

Ниже представлено решение на языке Python3

```python
n, m = [int(i) for i in input().split()]

better = [[] for i in range(int(1e6) + 1)]

for i in range(m):
    film1, film2 = [int(i) for i in input().split()]
    better[film2].append(film1)

for i in range(n):
    film1, film2, film3 = [int(i) for i in input().split()]
    ok = len(better[film1]) == 0
    for f in better[film2]:
        if not f == film1:
            ok = False
            break
    for f in better[film3]:
        if (not f == film1) and (not f == film2):
            ok = False
            break
    print("honest" if ok else "liar")
```

Задача I.1.1.5. Циклические сдвиги vs разворот строки (20 баллов)

На этой неделе на уроках информатики Васе рассказывают про строки.

Вчера Вася узнал, что такое циклический сдвиг:

k-й циклический сдвиг строки – это строка, полученная перестановкой первых k символов строки в её конец. В частности, 0-й циклический сдвиг строки – это сама строка.

И он написал программу, которая умеет перемещать первый символ строки в её конец k раз, получая таким образом k-й циклический сдвиг строки.

Сегодня Васе нужно реализовать разворот строки. Но у Васи тренировка. Ему некогда писать новые сложные программы. Поэтому он задался вопросом:

Можно ли циклическими сдвигами развернуть строку?

Благодаря своим друзьям Вася узнал, на какой строке (да, всего одной) будет тестировать его программу учитель, у которого нет времени рецензировать код каждого ученика. Поэтому вопрос упростился:

Можно ли циклическими сдвигами развернуть строку s?

Помогите ему ответить на этот вопрос.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 3 \cdot 10^5$) – длина строки s.

Во второй строке – сама строка s, на которой будет тестировать программу Васи учитель. Состоит строка s только из строчных латинских букв.

Формат выходных данных

Если строку нельзя развернуть циклическими сдвигами, то выведите число -1. В противном случае выведите такое целое число k ($0 \leqslant k < n$), что k-й циклический сдвиг строки s равен развёрнутой строке s. Если таких k несколько, выведите любое из них.

Пояснения

0-й циклический сдвиг строки s равен $abac$.

1-й циклический сдвиг строки s равен $baca$.

2-й циклический сдвиг строки s равен $acab$.

3-й циклический сдвиг строки s равен $caba$.

Развёрнутая строка s равна $caba$.

Единственное подходящее k равно трём.

Примеры

Пример №1

Стандартный ввод
4 abac
Стандартный вывод
3

Решение

Условие задачи можно свести к одному вопросу: можно ли циклическими сдвигами развернуть строку s?

Для нахождения ответа можно было искать развёрнутую строку s в строке $t = s + s$, так как все различные циклические сдвиги строки s являются подстроками строки t.

Наивная реализация этого поиска имеет асимптотику $O(n^2)$ по времени. За $O(n)$ по времени данную задачу можно было решить, например, с помощью алгоритма Кнута-Морриса-Пратта.

Другой способ решения: проверим, что строку s можно разрезать на два палиндрома. Быстро сделать это можно было, например, с помощью алгоритма Манакера.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
def pi(s):
    n = len(s)
    pi = [0] * n
    j = 0
    for i in range(1, n):
        while j > 0 and s[i] != s[j]:
            j = pi[j - 1]
        if s[i] == s[j]:
            j += 1
        pi[i] = j
    return pi


n = int(input())
s = input()

pi = pi(s[::-1] + '#' + s + s)
for i in range(n):
    if pi[n * 2 + i] == n:
        print(i)
        break
else:
    print(-1)
```

Вторая попытка

Задача I.1.2.1. Фанатам стратегий 2 (16 баллов)

В текущей миссии, очевидно, необходимо укрепить базу, прежде чем идти в открытый бой.

Для обеспечения устойчивой обороны требуется построить n различных новых зданий. Но не всё так просто.

Для поддержания процессов, которые будут происходить в этих зданиях, необходимо электричество. А получать электроэнергию новые здания могут только от новых электростанций. Новых электростанций на базе нет, так что их тоже придётся построить.

Зная, сколько единиц электроэнергии в единицу времени производит одна новая электростанция и количество электроэнергии, потребляемое за единицу времени каждым из упомянутых выше n новых зданий, определите минимальное количество электростанций, которое необходимо для полного функционирования требуемых n зданий.

Примечание: Считается, что электростанции не потребляют электроэнергии и среди n различных новых зданий, которые требуется построить, нет электростанции.

Формат входных данных

В первой строке заданы числа n и e ($1 \leqslant n \leqslant 10^5, 1 \leqslant e \leqslant 10^9$) – количество требуемых зданий и количество единиц электроэнергии, которое производит одна новая электростанция.

Во второй строке даны n чисел – количество единиц электроэнергии, потребляемое за единицу времени каждым из зданий. Все числа во второй строке неотрицательны и не превышают 10^9.

Формат выходных данных

Выведите одно целое число – минимальное количество электростанций, которое необходимо для полного функционирования требуемых n зданий.

Пояснения к примеру

Двух электростанции явно мало. Электроэнергию трёх электростанций можно распределить по зданиям следующим образом: на первое здание идёт 5 единиц от первой электростанции, на второе – 8 единиц от второй электростанции и 4 единицы от третьей, а на третье – 3 единицы от первой электростанции и 4 единицы от третьей.

Примеры

Пример №1

Стандартный ввод
3 8 5 12 7

Стандартный вывод
3

Решение

Пусть sum – суммарное количество единиц электроэнергии, потребляемое за единицу времени описанными n зданиями. Тогда, чтобы эти здания работали, нужно построить $\lceil \frac{sum}{s} \rceil$ новых электростанций.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n, s = [int(i) for i in input().split()]
sm = sum(int(i) for i in input().split())
print((sm + s - 1) // s)
```

Задача I.1.2.2. I Don't Like (20 баллов)

Кому-то не нравятся наши задачи. Наверно, из-за их сложности. Кто-то ругает нас за то, что программа не компилируется на компиляторах, имеющихся на Stepik, или просто не проходит наши тесты к задаче, хотя у кого-то на компьютере всё работает. Кто-то, не указывая на недочёты в задаче, хочет, чтобы ему или ей разжевали условие задачи, и после возмущается, прочитав, что мы не делаем пояснений и кратких пересказов условий, так как некорректностей найдено не было и мы хотим оставить всех участников олимпиады в равных условиях. А кто-то считает, что его тесты к задаче не хуже тех, что создали мы, и его решение верно, так как на его тестах оно работает (да, и такие есть). А кто-то списывает.

Всем этим замечательным людям мы можем лишь пожелать здоровья и бесконечного количества нервных клеток. Смириться с правилами олимпиады тоже не помешает.

А маленькому Коле не нравится, когда числа в массиве не отсортированы по возрастанию (если быть точным, по неубыванию, но Коля таких слов не знает).

Вот кто придумал дарить детям неотсортированные массивы? Мы не знаем, но Коля сегодня получил именно такой подарок. Он даже решил посчитать число таких пар индексов массива (i, j), что $i < j$ и $a_i > a_j$, чтобы хоть как-то измерить силу своей ненависти к подаренному ему массиву a и тому человеку, который это сделал.

Коля устал злиться, но сумеет сделать ещё ровно одно действие – поменять два элемента массива a местами. Ручки у него короткие, так что Коля может поменять местами только соседние элементы массива a (то есть такие элементы, индексы которых различаются не более чем на 1).

Определите количество способов, которыми Коля может уменьшить описанное выше число пар индексов. Два способа считаются различными, если существует индекс, который встречается только в одной из двух пар индексов, описывающих эти два способа.

Формат входных данных

В первой строке задано число n $(1 \leqslant n \leqslant 10^5)$ – количество элементов в массиве a.

Во второй строке даны n чисел a_i $(-10^9 \leqslant a_i \leqslant 10^9)$ – элементы массива a.

Гарантируется, что числа в массиве a не упорядочены по неубыванию.

Формат выходных данных

Выведите одно целое число – количество способов, которыми Коля может уменьшить описанное выше число пар индексов.

Примеры

Пример №1

Стандартный ввод
3 1 3 2

Стандартный вывод
1

Решение

Как изменится описанное в условии число пар индексов, если мы поменяем местами элементы a_i и a_{i+1} $(1 \leqslant i < n)$? Если $a_i > a_{i+1}$, то оно уменьшится на единицу; если $a_i < a_{i+1}$, то – увеличится на единицу; если $a_i = a_{i+1}$, то – не изменится.

Таким образом, в данной задаче достаточно посчитать количество пар соседних элементов в массиве a, в которых $a_i > a_{i+1}$.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
n = int(input())
a = [int(i) for i in input().split()]
print(sum(1 if a[i] > a[i + 1] else 0 for i in range(n - 1)))
```

Задача I.1.2.3. Мультимножества (20 баллов)

Дан набор из n чисел. Каждое число отнесли ровно к одному из 5-и мультимножеств: A, B, C, D или E.

По итогу такого распределения чисел получилось так, что все 5 мультимножеств непусты, суммы элементов мультимножеств равны и соблюдается следующее условие:

Для любых $a \in A$, $b \in B$, $c \in C$, $d \in D$ и $e \in E$ выполняется неравенство $a \leqslant b \leqslant c \leqslant d \leqslant e$.

Определите, правда ли, что такое могло произойти.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 10^5$) – размер набора чисел.

Вторая строка содержит n целых чисел a_i ($-10^9 \leqslant a_i \leqslant 10^9$) – сами числа набора.

Формат выходных данных

Выведите Yes, если возможно разбиение данных n чисел на мультимножества. Иначе выведите No.

Примеры

Пример №1

Стандартный ввод
19 2 1 1 2 2 0 2 3 11 3 3 4 3 4 0 6 5 1 2

Стандартный вывод
Yes

Решение

Пусть sum – сумма всех чисел данного набора, а $\max(X)$ и $\min(X)$ – наибольший и наименьший элементы множества X соответственно.

Тогда сумма элементов каждого из мультимножеств в отдельности равна $sum/5$ и описанное условие эквивалентно следующему:

$$\max(A) \leqslant \min(B) \leqslant \max(B) \leqslant \min(C) \leqslant \max(C) \leqslant \min(D) \leqslant \max(D) \leqslant \min(E)$$

Следовательно, для нахождения возможных мультимножеств стоит жадно добавлять наименьшие числа из набора в мультимножество A, пока сумма его элементов не достигнет $sum/5$, потом в мультимножество B, пока сумма его элементов не достигнет $sum/5$, и так далее. Если получилось сформировать мультимножества, удовлетворяющие всем условиям, то ответ `Yes`, иначе – `No`.

Асимптотика: $O(n \cdot \log(n))$

Пример программы-решения

Ниже представлено решение на языке Python3

```
n = int(input())
a = [int(i) for i in input().split()]
```

```
a.sort()
sum_a = sum(a)

if sum_a % 5 != 0:
    print('No')
    exit(0)

cur, cnt = 0, 1
for i in a:
    cur += i
    if cur == sum_a // 5 * cnt:
        cnt += 1
        if cnt == 6:
            break

print('Yes' if cnt == 6 else 'No')
```

Задача I.1.2.4. Фанатам стратегий 3 (24 баллов)

Данная задача – логическое продолжение задачи "Фанатам стратегий 2". Рекомендуем перед решением данной задачи полностью решить задачу "Фанатам стратегий 2".

Вскоре стало понятно, что всё совсем не просто. Нельзя взять и построить здание. Их в этой игре ещё и открыть нужно.

Новое здание типа A можно построить, только если на нашей базе функционирует хотя бы по одному новому зданию из списка необходимых зданий здания типа A.

Сколько на самом деле нам придётся построить зданий (не считая электростанций)? Какие они? В каком порядке их строить? Ваша задача – найти ответы на эти вопросы.

Примеры

Гарантируется, что существует такая последовательность постройки зданий, что здания всех типов можно построить.

Формат входных данных

В первой строке записаны три целых числа n, m и t ($1 \leqslant m \leqslant n \leqslant 5 \cdot 10^4$; $1 \leqslant t \leqslant 2$) – количество различных типов новых зданий в игре, количество новых зданий, которые нужно построить, и номер формата выходных данных.

В следующей строке записаны m названий типов зданий, разделённых пробелами – требуемые для обеспечения устойчивой обороны здания. Гарантируется, что строка не содержит одинаковых типов зданий.

Далее идёт n блоков по 2 строки следующего вида:

В первой строке – название типа здания.

Во второй – длина списка необходимых зданий для здания данного типа и сам список необходимых зданий. Гарантируется, что список не содержит одинаковых типов зданий.

Сумма длин списков необходимых зданий не превышает $5 \cdot 10^4$.

Название каждого типа здания состоит только из латинских букв и имеет длину не более десяти символов.

Формат выходных данных

Если $t = 1$, то выведите одно число – минимальное количество зданий, которые нужно построить.

Если $t = 2$, то в первой строке выведите одно число – минимальное количество зданий, которое необходимо построить, а во второй – k названий зданий, которые нужно построить, в том порядке, в котором их нужно строить. Если существует несколько подходящих последовательностей – выведите любую из них.

Примеры

Пример №1

Стандартный ввод

```
13 5 2
refinery vehicle repair palace turret
constryard
0
windtrap
1 constryard
refinery
1 windtrap
outpost
1 windtrap
silo
2 refinery constryard
vehicle
3 refinery windtrap outpost
barracks
2 constryard outpost
wall
1 outpost
turret
1 outpost
starport
2 silo refinery
repair
1 vehicle
hitech
3 vehicle wall outpost
palace
1 starport
```

Стандартный вывод

```
10
constryard windtrap refinery outpost silo vehicle turret starport repair
palace
```

Решение

Если представить, что здание типа A – это вершина орграфа, а список необходимых зданий здания типа A – список рёбер, направленных в вершину-здание типа A из других вершин-зданий, то задачу можно свести к задаче о нахождении топологической сортировки, с той лишь разницей, что вывести требуется не все вершины орграфа, а только те, из которых доступна хотя бы одна из m выделенных вершин-зданий.

Вершины-здания, которые нужно выводить, можно определить, например, с помощью серии обходов в глубину из m выделенных вершин-зданий, предварительно развернув в орграфе все рёбра. Порядок, в котором можно выводить эти вершины, может задать топологическая сортировка этих вершин. Топологическую сортировку можно найти, например, с помощь алгоритма Кана или алгоритма Тарьяна.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n, m, tp = [int(i) for i in input().split()]
need = set(input().split())
lists = {}
rev_lists = {}

for i in range(n):
    s = input()
    lists.setdefault(s, [])
    rev_lists.setdefault(s, [])
    for t in input().split()[1:]:
        lists.setdefault(t, [])
        lists[t].append(s)
        rev_lists[s].append(t)

q = [""] * n
ql = qr = 0
for s in lists.keys():
    if len(rev_lists[s]) == 0:
        q[qr] = s
        qr += 1

cnt = {}
while ql < qr:
    s = q[ql]
    ql += 1
    for t in lists[s]:
        cnt.setdefault(t, 0)
        cnt[t] += 1
        if cnt[t] == len(rev_lists[t]):
            q[qr] = t
            qr += 1

for s in reversed(q):
    if s in need:
        for t in rev_lists[s]:
            need.add(t)

ans = []
```

```python
for s in q:
    if s in need:
        ans.append(s)

print(len(ans))
if tp == 2:
    print(' '.join(map(str, ans)))
```

Задача I.1.2.5. Тир (20 баллов)

Давно были в тире? Мы недавно.

В нашем тире висят и стоят жестяные и алюминиевые банки из под различных напитков. Точнее, висели и стояли.

От наших выстрелов банки мотались из стороны в сторону на верёвке, срывались, звенели, мялись. Это вам не из пальцев стрелять.

Каждая из пуль либо прошла насквозь одной из банок, после чего поражённая банка упала на пол и откатилась в сторону так, что в неё было уже невозможно попасть; либо не попала ни в одну из банок. В любом случае, каждая из пуль застряла в стене, стоящей позади наших банок-мишеней.

Но тот день в прошлом. Осталась только стена с застрявшими в ней пулями и фотография. В попытке восстановить тот день и насладиться им снова мы собрали данные о положении каждой пули в стене, расположении банок и порядке выстрелов.

Помогите определить про каждую пулю, поразила ли она какую-то из банок, и если поразила, то какую именно.

Формат входных данных

В первой строке записаны два целых числа n и m $(1 \leqslant m,\ n \leqslant 10^5)$ – количество банок, которые были нашей мишенью в тот день, и количество совершённых в тот день выстрелов.

В i-ой из следующих n строк описывается положение i-ой банки. Положение задаётся координатами проекции банки на вертикальную плоскость. Проекция представляет из себя прямоугольник, стороны которого параллельны нанесённой на эту плоскость системы координат. Ось Y этой системы направлена вертикально вверх, а ось X – горизонтально. А прямоугольник задаётся парой точек – своей левой нижней и правой верхней вершинами.

Гарантируется, что ни одна пара этих прямоугольников не имеет ни одной общей точки.

В i-ой из следующих m строк описывается положение i-ой пули в стене. Пули заданы в том же порядке, в котором они выходили из наших дул. Сама стена строго вертикальна, поэтому мы можем считать, что положение задаётся координатами проекции пуль на вертикальную плоскость. Причём траектории движения пуль были строго перпендикулярна этой плоскости. Сами точки задаются парой координат в уже описанной выше системе координат.

Расстояние между банками и стеной по сравнению с расстоянием до стреляющих настолько мало, что мы им пренебрегаем.

Примечание: Значения координат по модулю не превышают 10^9.

Формат выходных данных

В первой и единственной строке выведите m чисел, i-ое из которых говорит, какую из банок i-я пуля прошла насквозь, если такая имеется. Если i не задела ни одну банку, то выведите -1, иначе выведите порядковый номер во входных данных банки, которую поразила i-я пуля.

Примеры

Пример №1

Стандартный ввод

```
4 10
0 0 1 1
2 3 3 8
15 15 20 20
10 12 12 13
2 2
0 -1
23 18
13 12
10 13
16 16
17 17
3 5
3 5
3 3
```

Стандартный вывод

```
-1 -1 -1 -1 4 3 -1 2 -1 -1
```

Решение

Один из способов решения:

Разобьём каждый прямоугольник на вертикальные отрезки: открывающий (с меньшей x координатой) и закрывающий (с большей x координатой).

Получившиеся отрезки отсортируем и будем обходить по неубыванию x координаты («запустим вертикальную сканирующую прямую от $-\infty$ до $+\infty$»). При равенстве x координаты сначала будем обрабатывать открывающие отрезки.

При обработке открывающего отрезка мы добавляем в некоторую структуру информацию о том, что сканирующая прямая на данный момент пересекает соответствующий отрезку прямоугольник. При обработке закрывающего отрезка мы удаляем из той же структуры информацию о том, что сканирующая прямая на данный момент пересекает соответствующий отрезку прямоугольник.

Так как прямоугольники не имеют общих точек, при любом положении сканирующей прямой открывающие отрезки, которые ещё «не закрыли», будут образовывать множество непересекающихся отрезков.

Следовательно можно множество открытых отрезков поддерживать, например, в set'е.

Если отсортировать точки-пули по неубыванию x координаты и параллельно обходить и их тоже, то мы сможем для каждой точки однозначно определить соответствующий набор открывающих отрезков; остаётся лишь найти ближайший к этой точке по y координате отрезок и проверить, лежит ли по y координате эта точка на нём. Для быстрого определения этого в set'е имеет смысл хранить открывающие отрезки в порядке, например, возрастания их «нижних» y координат и пользоваться функцией `lower_bound`.

Однако, так как мы изменили изначальный порядок точек-пуль, в случае если мы нашли для точки-пули прямоугольник, в который она попадает, это не значит, что данная пуля поразила соответствующую этому прямоугольнику банку. В проекцию банки могли попасть несколько проекций пуль, но только пуля, которая была выпущена раньше других, поразила банку.

Давайте для каждой проекции банки сохраним список проекций пуль, которые в него попадают. Тогда после окончания движения сканирующей прямой мы сможем определить для каждой банки, поразила ли её какая-либо пуля, и если поразила, то какая именно. А по этим данным мы можем восстановить ответ на задачу.

Асимптотика: $O(n \cdot \log(n))$

Пример программы-решения

Ниже представлено решение на языке C++

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

using namespace std;

typedef long long ll;

#define P pair
#define fi first
#define se second

#define V vector
typedef V<int> Vi;
typedef V<ll> Vll;

#define all(v) (v).begin(), (v).end()

#define forn(i, n) for (int (i) = 0; (i) < (n); (i)++)

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    int n, m;
    cin >> n >> m;
    Vll leftX(n), downY(n), rightX(n), upY(n);
    forn(i, n) {
        cin >> leftX[i] >> downY[i] >> rightX[i] >> upY[i];
    }
    Vll x(m), y(m);
```

```
    forn(i, m) {
        cin >> x[i] >> y[i];
    }

    V<P<ll, int>> s(n * 2 + m);
    forn(i, n) {
        s[i * 2] = {leftX[i], -i - 1};
        s[i * 2 + 1] = {rightX[i], m + i};
    }
    forn(i, m) {
        s[n * 2 + i] = {x[i], i};
    }
    sort(all(s));

    Vi killed_by(n);
    fill(all(killed_by), INT_MAX);

    map<ll, int> mp;

    forn(i, n * 2 + m) {
        if (s[i].se < 0) {
            mp[downY[-s[i].se - 1]] = -s[i].se - 1;
        } else if (s[i].se >= m) {
            mp.erase(downY[s[i].se - m]);
        } else if (!mp.empty() && mp.upper_bound(y[s[i].se]) != mp.begin() &&
                    y[s[i].se] <= upY[(--mp.upper_bound(y[s[i].se]))->se]) {
            killed_by[(--mp.upper_bound(y[s[i].se]))->se] =
                min(killed_by[(--mp.upper_bound(y[s[i].se]))->se], s[i].se);
        }
    }

    Vi ans(m);
    fill(all(ans), -1);
    forn(i, n) {
        if (killed_by[i] < INT_MAX) {
            ans[killed_by[i]] = i + 1;
        }
    }
    forn(i, m) {
        cout << ans[i] << ' ';
    }

    return 0;
}
```

Задачи первого этапа. Математика

Первая попытка. Задачи 8-9 класса

Задача I.2.1.1. (10 баллов)

Вася взял число 5689756193846349 и вычеркнул из него 8 цифр. В результате у него получилось максимально возможное число, которое таким образом можно получить из исходного. В ответ укажите это число.

Решение

Нужно вычеркнуть 8 цифр, чтобы число получилось максимально возможным. Для этого вы стараемся, использую оставшиеся зачеркивания, добиться того, чтобы на текущем месте стояла максимальная цифра, начиная с первого места.

Ход решения: у нас есть 8 зачеркивания и нужно, чтобы первая цифра была максимально возможной. Легко можно видеть, что используя 3 зачеркивания можно получить 9. Теперь у нас осталось 5 зачеркиваний и нужно, чтобы вторая цифра была максимально возможной. Видно, что, используя 4 зачеркивания, можно и на втором месте получить 9. У нас осталось одно зачеркивание и нужно выбрать цифру на третьем месте. Используя одно зачеркивание мы можем получить 8. Зачеркивания кончились.

Ответ: 99846349.

Задача I.2.1.2. (10 баллов)

На парковке 5×4 метра электрику нужно установить фонари так, чтобы в любом квадрате 2×2 метра был как минимум один 1 фонарь. Сколько минимально фонарей понадобится электрику?

Решение

Рассмотрим квадрат 3×3. Если поставить в центр фонарь, то в какую сторону не направить квадрат 2×2, он всегда будет с 1 фонарем, а это значит что фонари нам нужно ставить на расстоянии в 1 метр. Поставим 1 фонарь на расстоянии в 1 метр от верхней и левой сторон. Ставим 2 фонарь на расстояние в 1 метр от левой стены и 1 фонаря. 3 фонарь ставим на расстояние в 1 метр от нижней и правой сторон. Ставим 4 фонарь на расстояние в 1 метр от правой стены и 3 фонаря.

Ответ: 4.

Задача I.2.1.3. (10 баллов)

Настя наугад приписала к числу 375 справа 3 цифры. Число, которое у нее получилось, оказалось наибольшим из всех чисел, которые начинаются на 375 и делятся без остатка на 5, 6 и 7. В ответ укажите это число.

Решение

Полученное число должно делиться на $5\cdot6\cdot7 = 210$. И должно быть максимально возможным. Возьмем самое большое число из возможных: 375999 и поделим его на 210. Тогда $375999/210 = 1790 +$ остаток. Возьмем целую часть и умножим на 210. $1790 \cdot 210 = 375900$. Это и есть ответ.

Ответ: 375900.

Задача I.2.1.4. (10 баллов)

Однажды Ване понадобилось посчитать сумму натуральных чисел, состоящих только из цифр 1, 2, 3, 4, 5 без повторов (число не обязательно должно состоять из всех этих цифр, например, число 5 тоже подходит). Какое число получилось у Вани, если он посчитал всё верно?

Решение

Всего таких чисел будет $5! = 125$. При этом каждая цифра, в каждом разряде, будет стоять 25 раз (125/5). Тогда сумму можно представить как:

$$25 \cdot (10000 \cdot (1 + 2 + 3 + 4 + 5) + 1000 \cdot (1 + 2 + 3 + 4 + 5) + 100 \cdot (1 + 2 + 3 + 4 + 5)+$$

$$+10 \cdot (1 + 2 + 3 + 4 + 5) + (1 + 2 + 3 + 4 + 5)) = 4166625$$

Ответ: 4166625.

Задача I.2.1.5. (10 баллов)

Такси приедет к аэропорту не раньше чем в **15:25** и не позже чем в **15:50**. Для того, чтобы зарегистрироваться на самолёт и дойти до выхода на посадку, необходимо **20** минут. По расписанию выход закрывается в **15:50**, но если пассажир опаздывает, его ждут **10** минут, после чего закрывают выход и улетают. Найдите вероятность того, что человек, севший на это такси, успеет на самолёт. Ответ укажите в процентах.

Решение

Человек пройдёт регистрацию в 15:45 - 16:10 в зависимости от времени прибытия такси. С учётом времени ожидания самолёт вылетает в 16:00.

В промежутке от 15:45 до 16:00 – 15 минут, а в промежутке от 15:45 до 16:10 – 25 минут. Следовательно вероятностью того, что пассажир попадет на самолёт $15/25 = 0.6$.

Так как ответ нужно указать в процентах, ответ 60.

Ответ: 60.

Задача I.2.1.6. (10 баллов)

Предположим, что мы разматываем нить вдоль экватора Земли, а потом делаем то же самое, но в метре от экватора. Представим теперь, что мы разматываем один клубок вдоль поверхности Солнца и в метре от его поверхности, а второй – вдоль экватора Земли и в метре от его поверхности. К какому клубку нужно добавить больше ниток?

1. К тому, что мы разматываем в метре от Солнца

2. Одинаково
3. К тому, что мы разматываем в метре от Земли

Укажите ответ одним словом.

Решение

Длина окружности вычисляется по формуле $l = 2\pi R$, где R – радиус окружности. Если радиус увеличить на 1м, то длина дуги будет равна $L = 2\pi(R+1) = 2\pi R + 2\pi$, следовательно, не зависимо от радиуса, длина дуги в обоих случаях увеличится на одну и ту же величину: 2π, то есть примерно на 6 м.

Ответ: Одинаково.

Задача I.2.1.7. (10 баллов)

Количество пользователей системы «Умный дом», выпущенной компанией «Технологии будущего», росло в течение всего года. На четыре разных квартала (в какомто порядке) пришлись: наибольший абсолютный прирост, наименьший абсолютный прирост, наибольший относительный прирост и наименьший относительный прирост. (Абсолютный прирост – разность между новым и старым значением величины. Относительный прирост – это абсолютный прирост, делённый на старое значение.)

Известно, что наименьший относительный прирост был раньше, чем наибольший относительный. В каком квартале был наибольший абсолютный прирост? В ответ укажите номер квартала.

Решение

Способ 1. Докажем, что если количество пользователей растет и относительный прирост увеличивается, то увеличивается и абсолютный прирост. Пусть A и B – количество пользователей в какие-то моменты времени, причем $A < B$, а абсолютный прирост составляет x и y человек соответственно. Тогда относительный прирост равен соответственно $\frac{x}{A}$ и $\frac{y}{B}$. Если $\frac{x}{A} < \frac{y}{B}$, то $Bx < Ay < By$, следовательно, $x < y$.

Таким образом, наибольший относительный прирост не мог быть позже, чем наибольший абсолютный, и потому был не позже третьего квартала. Аналогично, наименьший относительный прирост не мог быть раньше, чем наименьший абсолютный прирост, и потому был не раньше второго квартала. Так как по условию задачи, наименьший относительный прирост был раньше, чем наибольший относительный, то они были во втором и третьем квартале соответственно. Следовательно, наибольший абсолютный прирост был позже, то есть в четвертом квартале.

Способ 2. Количество пользователей растет. Значит, если относительный прирост остается постоянным, то в следующем квартале он отсчитывается от большего значения, поэтому ему отвечает больший абсолютный прирост. Если же относительный прирост возрастает, то абсолютный прирост тем более возрастает.

Ответ: 4.

Задача I.2.1.8. (10 баллов)

Девять айтишников вошли в лифт на первом уровне одиннадцатиэтажного дворца техники. Если на одном из уровней вышли два человека, на другом – три, и еще на одном – четыре, то сколькими способами пассажиры могли выйти из лифта?

Решение

Распределить три группы на выход на трех уровнях из 10 можно $A_{10}^3 = \frac{10!}{(10-3)!}$ способами. Число перестановок из 9 человек с повторениями или число разбиений группы из девяти на три группы по 2,3 и 4 человека равно $N_9(2,3,4) = \frac{9!}{2!3!4!}$. Итого: $A_{10}^3 N_9(2,3,4) = \frac{10!}{4} = 907200$.

Ответ: 907200.

Задача I.2.1.9. (10 баллов)

В трех коробках лежат яблоки. Всего их 100. В первой коробке 50 яблок и все гнилые. Во второй коробке 30 яблок, десять из которых гнилые. А в третьей коробке 20 яблок, 5 из которых гнилые. Все яблоки из коробок переложили в пустой контейнер. Найдите вероятность того, что наугад выбранное яблоко из этого контейнера окажется не гнилым. Ответ укажите в виде десятичной дроби.

Решение

Длина окружности вычисляется по формуле $l = 2\pi R$, где R – радиус окружности. Если радиус увеличить на 1м, то длина дуги будет равна $L = 2\pi(R+1) = 2\pi R + 2\pi$, следовательно, не зависимо от радиуса, длина дуги в обоих случаях увеличится на одну и ту же величину: 2π, то есть примерно на 6 м.

Ответ: 0,35.

Задача I.2.1.10. (10 баллов)

Решите уравнение и запишите в качестве ответа разность наибольшего и наименьшего корня

$$x^2 + \frac{3x^2}{(x^2-2)} + 2 = 0.$$

Решение

Запомним, что модуль x не равен корню из двух. Домножим уравнение на $x^2 - 2$ и приведем подобные. Получим $x^4 + 3x^2 - 4 = 0$. Решим его как квадратное относительно x^2, получим $x^2 = 1$ и $x^2 = -4$. Так как корень числа не может быть отрицательным, остается только $x^2 = 1$. Возьмем корень справа и слева и получим, что $x = 1$ и $x = -1$. Вычтем $(1 - (-1) = 2)$.

Ответ: 2.

Первая попытка. Задачи 10-11 класса

Задача I.2.2.1. (10 баллов)

Предположим, что мы разматываем нить вдоль экватора Земли, а потом делаем то же самое, но в метре от экватора. Представим теперь, что мы разматываем один клубок вдоль поверхности Солнца и в метре от его поверхности, а второй – вдоль экватора Земли и в метре от его поверхности. К какому клубку нужно добавить больше ниток?

1. К тому, что мы разматываем в метре от Солнца
2. Одинаково
3. К тому, что мы разматываем в метре от Земли

Укажите ответ одним словом.

Решение

Длина окружности вычисляется по формуле $l = 2\pi R$, где R – радиус окружности. Если радиус увеличить на 1м, то длина дуги будет равна $L = 2\pi(R+1) = 2\pi R + 2\pi$, следовательно, не зависимо от радиуса, длина дуги в обоих случаях увеличится на одну и ту же величину: 2π, то есть примерно на 6 м.

Ответ: Одинаково.

Задача I.2.2.2. (10 баллов)

Девять айтишников вошли в лифт на первом уровне одиннадцатиэтажного дворца техники. Если на одном из уровней вышли два человека, на другом – три, и еще на одном – четыре, то сколькими способами пассажиры могли выйти из лифта?

Решение

Распределить три группы на выход на трех уровнях из 10 можно $A_{10}^3 = \frac{10!}{(10-3)!}$ способами. Число перестановок из 9 человек с повторениями или число разбиений группы из девяти на три группы по 2,3 и 4 человека равно $N_9(2,3,4) = \frac{9!}{2!3!4!}$. Итого: $A_{10}^3 N_9(2,3,4) = \frac{10!}{4} = 907200$.

Ответ: 907200.

Задача I.2.2.3. (10 баллов)

Вычислите: $\cos 36^\circ - \sin 18^\circ$.

Решение

Разделим и умножим данное выражение на одно и то же, не равное нулю, число, затем вопспользуемся формулами преобразования произведения тригонометрических функций в сумму:

$$\cos 36^\circ - \sin 18^\circ = \frac{\cos 36^\circ \cos 18^\circ - \sin 18^\circ \cos 18^\circ}{\cos 18^\circ} =$$
$$= \frac{0,5\cos 18^\circ + 0,5\cos 54^\circ - 0,5\sin 36^\circ}{\cos 18^\circ} = \frac{0,5\cos 18^\circ}{\cos 18^\circ} = 0,5$$

Ответ: 0,5.

Задача I.2.2.4. (10 баллов)

Компьютер последовательно решает несколько задач. Было замечено, что на решение каждой следующей задачи компьютер тратил в одно и то же число раз меньше времени, чем на решение предыдущей. Сколько было предложено задач, и сколько времени затрачено машиной на решение всех задач, если на решение всех задач, кроме первой, затрачено 63,5 мин.; на решение всех задач, кроме последней, затрачено 127 мин.; а на решение всех задач, кроме двух первых и двух последних, затрачено 30 мин?

Напишите ответ в формате: X задач(и), Y мин

Решение

Пусть b_i – время, затрачиваемое машиной на решение i-ой задачи – образуют геометрическую прогрессию со знаменателем $0 < q < 1$, то есть $b_i = b_1 q^{i-1}$. Тогда

$$b_2 + b_3 + b_4 + \cdots + b_n = 63,5,$$

$$b_1 + b_2 + b_3 + \cdots + b_{n-1} = 127,$$
$$b_3 + b_4 + \cdots + b_{n-2} = 30.$$

Анализируя цифры, видим, что q может быть только $\frac{1}{2}$. Рассмотрим третье равенство с разным количеством членов. Если в нем только один член $b_3 = 30$, тогда $b_1 q^2 = 30$, $b_1 = 120$, и мы имеем ряд $120+60+30+15+7,5$, не удовлетворяющий второму условию. Если в третьем уравнении два члена: $b_3 + b_4 = 30$, то $b_1 q^2(1+q) = 30$, и подставляя $q = \frac{1}{2}$, находим $b_1 = 80$. В таком случае имеем ряд 80+40+20+10+5+2,5, который опять не удовлетворяет второму уравнению. Если в третьем уравнении три члена: $b_3 + b_4 + b_5 = 30$, то из уравнения $b_1 q^2(1+q+q^2) = 30$ находим $b_1 = 30 \cdot 4 \cdot 4/7$ – не целое число, что не возможно. И наконец, если в третьем уравнении 4 члена: $b_3 + b_4 + b_5 + b_6 = 30$, то $b_1 = 64$, и получаем ряд $64 + 32 + 16 + 8 + 4 + 2 + 1 + 0,5$, который удовлетворяет всем условиям. Сумма этого ряда 127,5 мин.

Ответ: 8, 127,5.

Задача I.2.2.5. (10 баллов)

4 раза в сутки с интервалом 6 часов в порту с. Никольское (Камчатка) измеряют уровень воды. Со временем обнаружили, что колебания уровня воды приблизительно задаются уравнением

$$h = 1,4 - 0,8sin(0,3t)$$

Вопрос 1. Найдите высоту прилива через 15 часов с начала измерений.
Вопрос 2. В какое время ожидался самый высокий уровень воды в первые сутки с округлением до часов?

Решение

Вопрос 1. Для решения подставим значение $t = 15$ в формулу и проведем вычисления, считая значение синуса в радианах.

$$h = 1,4 - 0,8sin(0,3 \cdot 15)$$

$$h = 1,4 - 0,8 \cdot (-0,978)$$

$$h = 2,18m$$

Ответ: $h = 2,18m$.

Вопрос 2. Для поиска самого высокого уровня воды нужно вычислить экстремумы функции $h = 1,4 - 0,8sin(0,3t)$ для $t \in [0;24]$и определить точку максимума.

Известно, что в точках экстремума производная функции равна нулю, а при переходе через точку максимума знак производной меняется с положительного на отрицательный. Будем искать решение по шагам.

1. Найдем производную функции

$$h\prime = 1,4\prime - (0,8sin(0,3t))\prime$$

$$h\prime = -0,8(sin(0,3t))\prime$$

$$h\prime = -0,8cos(0,3t) \cdot (0,3t)\prime$$

$$h\prime = -0,8cos(0,3t) \cdot 0,3$$

$$h\prime = -2,4cos(0,3t)$$

2. Приравняем производную к нулю и решим полученное тригонометрическое уравнение

$$-2,4cos(0,3t) = 0$$

$$cos(0,3t) = 0$$

$$0,3t = \pm arccos(0) + 2\pi n$$

$$0,3t = \frac{\pi}{2} + \pi n$$

$$t = 1,67\pi + 3,33\pi n$$

Вычислим, при каких значениях n $t \in [0;24]$, получим два значения: $t_1 = 5,34$ и $t_2 = 15,79$

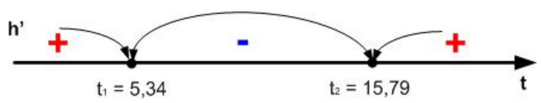


Рис. I.2.1: Знаки производной на промежутках

Отметим значения t_i на числовой прямой и проверим знаки производной на полученных промежутках (см. рис. I.2.1)

Ответ: 2,18, 16:00.

Задача I.2.2.6. (10 баллов)

К конгрессу математиков было принято решение оформить клумбу цветами разных видов так, как показано на рисунке I.2.2. Для заполнения треугольника ABC было использовано 72 цветка. Сколько цветков при той же плотности посадки понадобится для заполнения четырехугольника $BCDE$, если стороны AB и AC равны, соответственно, 9 и 8 метров, а диаметр круга равен 10 метрам. Отрезок AD проходит в точности через центр круга.

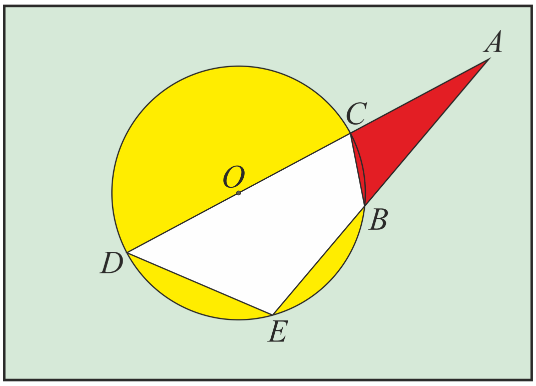


Рис. I.2.2: Чертёж клумбы

Решение

Произведения отрезков секущих, проведенных из общей точки, равны: $AC \cdot AD = AB \cdot AE$, а угол $\angle A$ является общим углом треугольников $\triangle ABC$ и $\triangle ADE$. Соответственно, данные треугольники подобны. Коэффициент подобия равен отношению сторон AD и AB: $k = \frac{AD}{AB} = \frac{(8+10)}{9} = 2$, т.к. отрезок DC совпадает с диаметром круга. Значит, площади треугольников $\triangle ADE$ и $\triangle ABC$ соотносятся как квадрат коэффициента подобия: $S_{\triangle ADE} = k^2 \cdot S_{\triangle ABC} = 4S_{\triangle ABC}$.

Далее имеем: $S_{BCDE} = S_{\triangle ADE} - S_{\triangle ABC} = 3S_{\triangle ABC}$

Так как плотность посадки цветов одинаковая, то для заполнения четырехугольника $BCDE$ необходимо $N = 3 \cdot 72 = 216$ цветков.

Ответ: 216 цветков.

Задача I.2.2.7. (10 баллов)

Дизайнер в известной компании придумал новый продукт для поддержания имиджа – термос-кружку с логотипом компании. Внутренняя поверхность термоса имеет форму прямого усеченного конуса с диаметрами нижнего и верхнего оснований 10 см и 8 см соответственно. В нижней части поверхность завершается шаровым сегментом высотой 2 см. Определите высоту термоса от макушки шарового сегмента до верхнего основания конуса, чтобы в него помещалась жидкость объемом 0,7 литра.

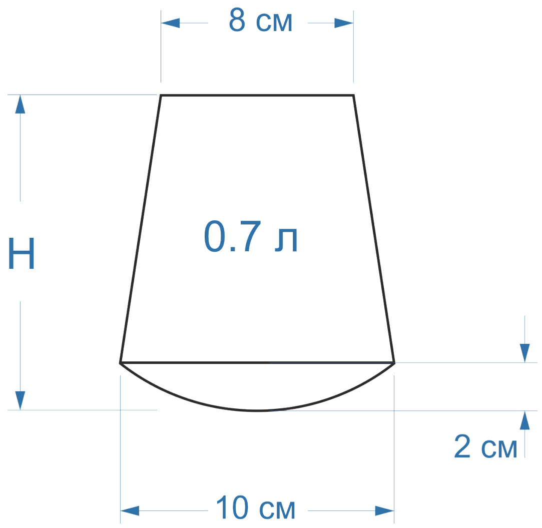


Рис. I.2.3: Схема внутренней поверхности термоса

Решение

Представим схему внутренней поверхности термоса, как показано на рисунке. H на рисунке I.2.3 - высота, которую нужно найти.

Для рассчета объема шарового сегмента все исходные данные в задаче имеются $V_1 = \frac{1}{6}\pi h_1(h_1^2 + 3r_{bot}^2) = \frac{1}{6}\pi 2(2^2 + 3 \cdot 5^2) = \frac{79}{3}\pi$. Объем усеченного конуса определяется выражением $V_2 = \frac{1}{3}h_2\pi(r_{bot}^2 + r_{bot} \cdot r_{top} + r_{top}^2) = \frac{1}{3}h_2\pi(5^2 + 5 \cdot 4 + 4^2) = \frac{61}{3}h_2\pi$. Общий объем термоса складывается из объема шарового сегмента и усеченного конуса $V = V_1 + V_2$. Подставим в эту формулу выражения для объемов шарового сегмента и конуса, а затем выразим высоту конуса через известный из условия задачи объем термоса, получим $h_2 = \frac{3(700 - \frac{79\pi}{3})}{61\pi} \approx 9.66$. Следовательно искомая высота термоса будет равна $H \approx 2 + 9.66 = 11.66$ см.

Ответ: 11.66 см.

Задача I.2.2.8. (10 баллов)

Известная строительная компания выпускает универсальные комплектующие для сооружения легких каркасных конструкций типа «пирамида». Отдельный элемент конструкции имеет форму равнобедренного треугольника с углом между боковыми сторонами 15 градусов. При монтаже конструкции треугольники соединяются попарно своими боковыми сторонами. Высота конструкции, составленной из 8 таких элементов, равна 4 метра. Какова будет высота конструкции, если количество элементов уменьшить до 3?

Решение

Согласно условию задачи, конструкция представляет собой правильную пирамиду в основании, которой лежит восьмиугольник.

Рассмотрим прямоугольный треугольник, образованный высотой пирамиды и высотой ее боковой стороны. Высоты пирамиды H и боковой грани h являются соответственно катетом и гипотенузой данного треугольника. Второй его катет можно выразить через длину основания боковой грани пирамиды a, пользуясь тем, что основанием пирамиды является правильный восьмиугольник: $OO' = a(\frac{1}{2} + \sqrt{2})$. По теореме Пифагора $OO'^2 + H^2 = h^2$. Комбинируя два эти выражения, получим соотношение между высотами и длиной основания $a^2(\frac{1}{2} + \sqrt{2})'^2 + H^2 = h^2$

Боковая грань пирамиды является равнобедренным треугольником. Его высоту можно выразить через длину основания и угол при боковых сторонах $h = \frac{a}{2\tan\frac{\phi}{2}}$. Подставляя данное выражение в предыдущее уравнение получим для длины основания $a = \frac{H}{\sqrt{\frac{1}{4\tan^2\frac{\phi}{2}} - (\frac{1}{2}+\sqrt{2})^2}}$, вычисляя которое, имеем $a \approx 1.22$ метра. Соответственно найдем $h \approx 4.63$ метра.

Теперь рассмотрим пирамиду, в основании которой лежит правильный треугольник. В прямоугольном треугольнике образованном высотами пирамиды H_1 и боковой грани h, катет в плоскости основания пирамиды равен $OO' = \frac{a}{2\sqrt{3}}$. Соотношение между высотами и длиной основания боковой грани запишем в виде $\frac{a^2}{12} + H_1^2 = h^2$, из которого найдем $H_1 \approx 4.62$ метра.

Ответ: 4.21 метра.

Задача I.2.2.9. (10 баллов)

Количество пользователей системы «Умный дом», выпущенной компанией «Технологии будущего», росло в течение всего года. На четыре разных квартала (в какомто порядке) пришлись: наибольший абсолютный прирост, наименьший абсолютный прирост, наибольший относительный прирост и наименьший относительный прирост. (Абсолютный прирост – разность между новым и старым значением величины. Относительный прирост – это абсолютный прирост, делённый на старое значение.)

Известно, что наименьший относительный прирост был раньше, чем наибольший относительный. В каком квартале был наибольший абсолютный прирост? В ответ укажите номер квартала.

Решение

Способ 1. Докажем, что если количество пользователей растет и относительный прирост увеличивается, то увеличивается и абсолютный прирост. Пусть А и В – количество пользователей в какие-то моменты времени, причем $A < B$, а абсолютный прирост составляет x и y человек соответственно. Тогда относительный прирост равен соответственно $\frac{x}{A}$ и $\frac{y}{B}$. Если $\frac{x}{A} < \frac{y}{B}$, то $Bx < Ay < By$, следовательно, $x < y$.

Таким образом, наибольший относительный прирост не мог быть позже, чем наибольший абсолютный, и потому был не позже третьего квартала. Аналогично, наименьший относительный прирост не мог быть раньше, чем наименьший абсолютный прирост, и потому был не раньше второго квартала. Так как по условию задачи, наименьший относительный прирост был раньше, чем наибольший относительный, то они были во втором и третьем квартале соответственно. Следовательно, наибольший абсолютный прирост был позже, то есть в четвертом квартале.

Способ 2. Количество пользователей растет. Значит, если относительный прирост остается постоянным, то в следующем квартале он отсчитывается от большего значения, поэтому ему отвечает больший абсолютный прирост. Если же относительный прирост возрастает, то абсолютный прирост тем более возрастает.

Ответ: 4.

Задача I.2.2.10. (10 баллов)

Имеется информационная сеть, состоящая из центров хранения информации. Некоторые пары центров соединены каналами связи. Обмен информацией между любыми двумя центрами выполняется либо непосредственно, либо через другие каналы и центры. Если каждая пара центров может обмениваться информацией, сеть является исправной.

Известно, что в сети всего $n = 12$ центров, каждый из центров непосредственно связан каналом с $k = 7$ другими центрами.

Какое наименьшее количество центров надо разрушить, чтобы сеть стала неисправной?

Решение

Информационной сети следует сопоставить граф G: верршины графа - это центры сети, ребра - каналы связи. Если сеть исправна, то ей соответствует связный граф. Разрушить сеть можно или удалением вершин (разрушением центров) или удалением ребер (повреждением каналов). В результате любого из выбранных способов должен получиться несвязный граф. В данной сети минимальная степень вершины $\delta(G) = 7$. Известно, что для графа, не являющегося полным и имеющего n вершин выполняется неравенство:

$$k(G) \geqslant 2\delta(G) - n + 2,$$

где $k(G)$ - связность графа G.

Подставим в формулу данные о нашей сети

$$k(G) \geqslant 2 \cdot 7 - 12 + 2$$

$$k(G) \geqslant 4$$

Из последнего неравенства следует, что если удалить из сети минимально возможное количество центров, равное 4, граф распадется на несвязные области.

Ответ: 4.

Вторая попытка. Задачи 8-9 класса

Задача I.2.3.1. (10 баллов)

Вася загадал какое-то трехзначное число. После этого Вася начал это число делить на 7 с остатком следующим образом: он поделил число на 7, записал его остаток и начал делить неполное частное от предыдущего деления.

Пример того, как Вася делил число 548:

$548/7 = 78$ остаток 2

$78/7 = 11$ остаток 1

$11/7 = 1$ остаток 4

В ответ напишите, какое число загадал Вася, если сумма его остатков наибольшая, а само число наименьшее из возможных для данной суммы.

Решение

Так как требуется минимальное трехзначное число, то найдем количество цифр 7, которые при перемножении получаем трехзначное число.

$$7^3 = 343$$

$$7^4 = 2401$$

Из этого следует, что необходимо сделать три перемножения для нахождения числа.

Максимальная сумма остатков является сумма максимальных остатков. Максимальный остаток - цифра 6, т.е. $6+6+6 = 18$. Для того, чтобы найти искомое число, необходимо к минимальному числу добавить остаток и умножить его на 7. Покажем пошагово:

$$7 + 6 = 13$$

$$13 \cdot 7 = 91$$

$$91 + 6 = 97$$

$$97 \cdot 7 = 679$$

$$679 + 6 = 685$$

Ответ: 685.

Задача I.2.3.2. (10 баллов)

Пете, чтобы добраться домой из деревни, вначале нужно сесть на автобус номер 35, затем на электричку, следующую по маршруту Васюки-Москва, а после этого сесть на автобус номер 465.

Автобус номер 35 едет до железнодорожной станции Васюки 45 минут и, в зависимости от пробок и настроек светофоров, может ехать на 9 минут дольше или на 5 минут быстрее. На поездку на электричке Петя тратит 70 ± 25 минут.

Петя сел на автобус номер 35 в **16:25**. Электрички по расписанию отправляются со станции Васюки с **7:15** каждые полчаса. А автобус 465 от железнодорожного вокзала в Москве с **8:30** каждые 45 минут.

Напишите время отправления автобуса 465 от вокзала в Москве, на который Петя сядет с наибольшей вероятностью. Пешие переходы между остановками автобусов и платформами электрички считать равными 0.

Ответ укажите в виде четырехзначного числа вида ччмм, где первые две цифры отвечают за часы, а оставшиеся — за минуты. Для времени **9:15** нужно написать 0915.

Решение

Рассмотрим варианты прибытия Пети до электрички на автобусе номер 35, время отбытия **16:25**. Автобус едет 45 минут, может ехать на 9 минут дольше или на 5 минут быстрее. Таком образом получаем три варианта:

А. 16:25 + 45 минут + 9 минут = 17:19

Б. 16:25 + 45 минут = 17:10

В. 16:25 + 40 минут = 17:05

Рассмотрим расписание электричек до Москвы: 7:15, 7:45, 8:15, 8:45, . . . , 17:15, 17:45 . . . Следовательно, при варианте А - Петя успевает на электричку в 17:45, при вариантах Б и В - Петя успевает на электричку в 17:15. Электричка едет 70 ± 25 минут, получаем:

А.1. 17:45 + (70 + 25) минут = 19:20

А.2. 17:45 + 70 минут = 18:55

А.1. 17:45 + (70 - 25) минут = 18:30

Б.1. 17:15 + (70 + 25) минут = 18:50

Б.2. 17:15 + 70 минут = 18:25

Б.3. 17:15 + (70 - 25) минут = 18:00

В.1. 17:15 + (70 + 25) минут = 18:50

В.2. 17:15 + 70 минут = 18:25

В.3. 17:15 + (70 - 25) минут = 18:00

Рассмотрим расписание автобуса 465: 8:30, 9:15, 10:00, 10:45, . . . , 17:30, 18:15, 19:00, 19:45 . . . Тогда:

А.1. 19:20 → 19:45

А.2. 18:55 → 19:00

А.1. 18:30 → 19:00

Б.1. 18:50 → 19:00

Б.2. 18:25 → 19:00

Б.3. 18:00 → 18:15

В.1. 18:50 → 19:00

В.2. 18:25 → 19:00

В.3. 18:00 → 18:15

С бóльшей вероятностью Петя успеет на атобус 465, который отправляется в

19:00.

Ответ: 1900.

Задача I.2.3.3. (10 баллов)

Решите уравнение: $\cos\sqrt{x} + \sqrt{\cos x} = 2$.

Решение

Так как $|\cos\sqrt{x}| \leqslant 1$ и $|\sqrt{\cos x}| \leqslant 1$, то уравнение $\cos\sqrt{x}+\sqrt{\cos x} = 2$ равносильно системе уравнений:

$$\begin{cases} \cos\sqrt{x} = 1, \\ \sqrt{\cos x} = 1. \end{cases}$$

Из певрого уравнения системы получим, что $x = 2\pi k$, где $k \in Z$, а из второго уравнения получим, что $\sqrt{x} = 2\pi n \Leftrightarrow x = 4\pi^2 n^2$, где $n \in Z$, $n \geqslant 0$. Найдем пересечение полученных множеств: $2\pi k = 4\pi^2 n^2 \Leftrightarrow k = 2\pi n^2$. Так как π - иррациональное число, то последнее равенство выполяется тогда, и только тогда, когда $n = k = 0$. Следовательно, единственным решением данного уравнения является $x = 0$.

Ответ: $x = 0$.

Задача I.2.3.4. (10 баллов)

На столе разложены 2020 спичечных коробков, в некоторых из них есть спички, а в некоторых — нет.

На первом коробке написано: «Все коробки пустые».

На втором — «По крайней мере 2019 коробков пустые».

На третьем — «По крайней мере 2018 коробков пустые».

На 2020-ом — «По крайней мере один коробок пустой».

Известно, что надписи на пустых коробках ложные, а на коробках со спичками — истинные. Определите, сколько коробков не пустые.

Решение

Рассмотрим коробок №1.

1. Если коробок пуст, то надпись на нем ложна.
2. Если коробок полон, то надпись на нем верна. Следовательно, все 2020 должны быть пусты, получаем противоречие.

Коробок №1 полон. Отсюда, надпись на коробке №2020 верна и он полон.

Рассмотрим коробок №2.

Мы уже знаем, что один коробок полон и один коробок пуст.

1. Если коробок пуст, то надпись на нем ложна.
2. Если коробок полон, то надпись на нем верна. Следовательно, 2019 коробков должны быть пусты, получаем противоречие, т.к. коробки №2 и №2020 полные.

Отсюда коробоки №1 и №2 пусты, коробки №2019 (надпись на коробке верна) и №2020 полны.

Рассмотрим коробок №3.

Мы уже знаем, что два коробка полоны и два коробка пусты.

1. Если коробок пуст, то надпись на нем ложна.
2. Если коробок полон, то надпись на нем верна. Следовательно, 2018 коробков должны быть пусты, получаем противоречие, т.к. коробки №3, №2019 и №2020 полные.

Отсюда коробоки №1, №2 и №3 пусты, коробки №2018 (надпись на коробке верна), №2019 и №2020 полны.

Продолжая рассуждения, понимаем, что ровно половина коробков пустые, половина коробков полные.

Ответ: 1010.

Задача I.2.3.5. (10 баллов)

Саша любит придумывать и записывать "волшебные" числа. Для него это пятизначные числа, на чётных разрядах которых находятся чётные цифры, а на нечётных — нечётные. Числа: 12301 и 36789 — "волшебные", а 78645 — нет.

Юноша занимается этим нелегким делом очень давно, и за это время он успел выписать все такие числа. Однажды он посмотрел на эти числа и решил сосчитать, сколько цифр "5" ему пришлось написать, для того, чтобы записать все "волшебные" числа по одному разу.

Сколько цифр "5" насчитал Саша, если известно, что он не ошибся?

Решение

"Волшебное" число выглядит так:

$$\mathbf{н} \,\|\, \mathbf{ч} \,\|\, \mathbf{н} \,\|\, \mathbf{ч} \,\|\, \mathbf{н}$$

Где **н** - нечетная цифра, **ч** - четная цифра

Всего четных цифр - 5: 0, 2, 4, 6, 8.

Всего нечетный цифр - 5: 1, 3, 5, 7, 9.

На каждой позиции может стоять по 5 чисел, следовательно всего таких чисел:

$$5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 = 3125$$

Есть несколько вариантов чисел, котрые включают цифру 5:

1.	5	**ч**	**н** - 5	**ч**	**н** - 5
2.	**н** - 5	**ч**	5	**ч**	**н** - 5
3.	**н** - 5	**ч**	**н** - 5	**ч**	5
4.	5	**ч**	5	**ч**	**н** - 5
5.	5	**ч**	**н** - 5	**ч**	5
6.	**н** - 5	**ч**	5	**ч**	5
7.	5	**ч**	5	**ч**	5

Где (**н** - 5) – нечетные числа, не включающие число 5: 1, 3, 7, 9.

Таким образом получаем, что чисел с тремя цифрами 5:

$$1 \cdot 5 \cdot 1 \cdot 5 \cdot 1 = 1200,$$

чисел с двумя цифрами 5:

$$(1 \cdot 5 \cdot 1 \cdot 5 \cdot 4) \cdot 3 = 300,$$

чисел с одной цифрой 5:

$$(1 \cdot 5 \cdot 4 \cdot 5 \cdot 4) \cdot 3 = 1200.$$

В задании спрашивается, сколько записей цифры 5 было, следовательно, необходимо полученные результаты умножить на количество 5 в записи чисел:

$$1200 \cdot 1 + 300 \cdot 2 + 25 \cdot 3 = 1875$$

Ответ: 1875.

Задача I.2.3.6. (10 баллов)

Двое друзей, каждый со своей позиции, ведут наблюдение через вертикальную щель в круглую комнату.

Определить величину щели, если вместе они контролируют только четвертую часть стены комнаты, и при этом угол зрения одного и второго равны соответственно °10 и °20 градусов. При решении считать, что каждый из них видит свой участок стены.

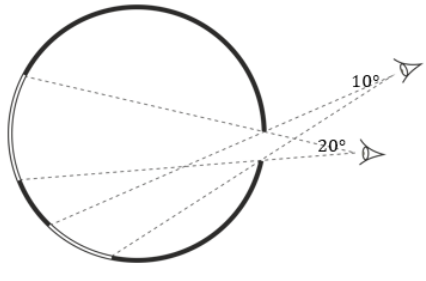


Рис. I.2.4: Схема задачи

Решение

Дополним схему задачи следующими обозначениями, как показано на рисунке. Общую дугу окружности - щель обозначим символом m, а дуги, показывающие сектора обзора, обозначим через n_1 и n_2. Углы обозначающие сектора обзора назовём $\angle A$ и $\angle B$ соответственно.

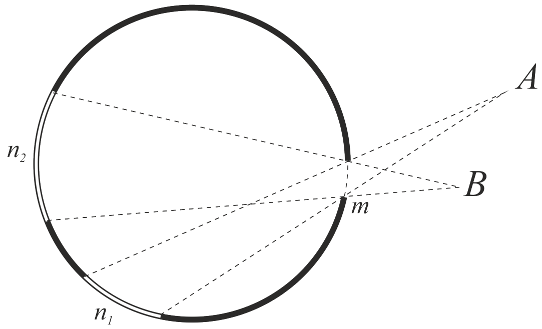


Рис. I.2.5: Схема задачи

Известно, что угол $\angle A$ между секущими, пересекающимися вне окружности, вычисляется по формуле: $\angle A = \frac{n_1 - m}{2}$. Аналогично, $\angle B = \frac{n_2 - m}{2}$. Также по условию задачи $n_1 + n_2 = 90°$. Тогда, складывая левые и правые части двух первых уравнений, получим выражение $\angle A + \angle B = \frac{n_1 + n_2}{2} - m$. Откуда можем найти искомую дугу $m = \frac{n_1 + n_2}{2} - \angle A - \angle B = \frac{90°}{2} - 10° - 20° = 15°$.

Ответ: °15.

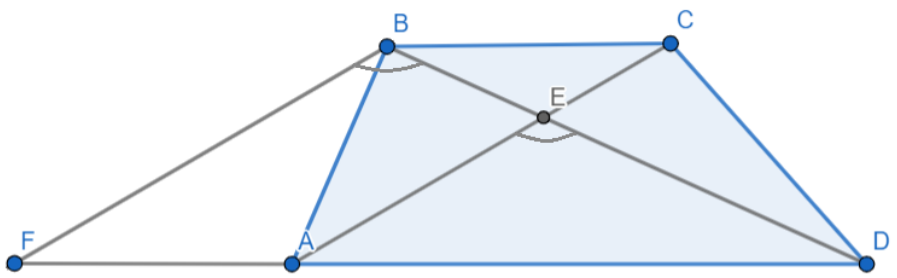
Задача I.2.3.7. (10 баллов)

В трапеции $ABCD$ с основаниями AD и BC диагонали пересекаются в точке E. $\angle AED = 60°$. $AC = BC + AD$. Найдите отношение $AB : CD$.

Введите ответ с точностью **не менее** 3 знаков после запятой (отличие от точного значения не должно превышать 0.0005).

Решение

Построим параллелограмм $AFCD$.



Тогда $\angle FBD = \angle AOD$, $AF = BC$ и

$$FB = AC = BC + AD = AF + AD = FD.$$

Значит, треугольник $\triangle FBD$ – равнобедренный ($FB = FD$). Поскольку углы при основании равнобедренного треугольника – острые, то угол $\angle FBD$ не может быть равен 120°. Следовательно, он равен 60°, и треугольник $\triangle FBD$ – равносторонний. Поэтому $BD = FB = AC$, то есть диагонали трапеции равны между собой. Отюда следует, что трапеция равнобедренная.

Таким образом $\frac{AB}{CD} = 1$.

Ответ: 1.

Задача I.2.3.8. (10 баллов)

Решите уравнение $-3x^5 + 375x^3 - 7500x = 0$.

В ответ запишите сумму наибольшего и наименьшего из неотрицательных корней.

Решение

Разделим уравнение на -3.

$$x^5 - 125x^3 + 2500x = 0$$

$$x(x^4 - 125x^2 + 2500) = 0$$

Разобьем слагаемое $-125x^2$ на два $-100x^2$ и $-25x^2$:

$$x(x^4 - 100x^2 - 25x^2 - 2500) = 0$$

$$x(x^2(x^2 - 100) - 25(x^2 - 100)) = 0$$

$$x(x^2 - 100)(x^2 - 25) = 0$$

Применим формулу разности квадратов:

$$x(x - 10)(x + 10)(x - 5)(x + 5) = 0$$

Корни уравнения: $x_1 = 0$, $x_2 = 5$, $x_3 = -5$, $x_4 = 10$, $x_5 = -10$.

Следовательно, минимальный корень - x_1 и максимальный x_4: $x_4 - x_1 = 10$.

Ответ: 10.

Задача I.2.3.9. (10 баллов)

Составляя расписание мастер-классов для финала олимпиалы НТИ по профилю "Разработка приложений виртуальной и дополненной реальности"организаторы столкнулись с затруднением. Мастер-классы должны проводится 1 раз в день с понедельника по четверг, но приглашенные специалисты имеют ряд требований к дню проведения мастер-класса. Предложите варианты расписания мастер-классов, если:

а) специалист по математическому моделированию может провести провести мастер-класс в понедельник, вторник или среду;

б) специалист по 3D моделированию может пообщаться с финалистами или в понедельник, или во вторник;

в) мастер-класс по физике может состояться во вторник или четверг;

г) мастер-класс по программированию интерактивного окружения - только во вторник или среду.

Сколько возможно вариантов составления расписания мастер-классов для проведения финала? Укажите их, перечисляя через точку с запятой мастер-классы в порядке их следования в течение финала. Например: Вариант 1. Математическое моделирование; 3D моделирование; программирование интерактивного окружения; физика.

Решение

Построим граф, вершины которого - возможные дни недели (1, 2, 3, 4) и мастер-классы (м, ф, п, 3d). Ребрами соединим мастер-классы с возможными днями проведения(рис. I.2.10).

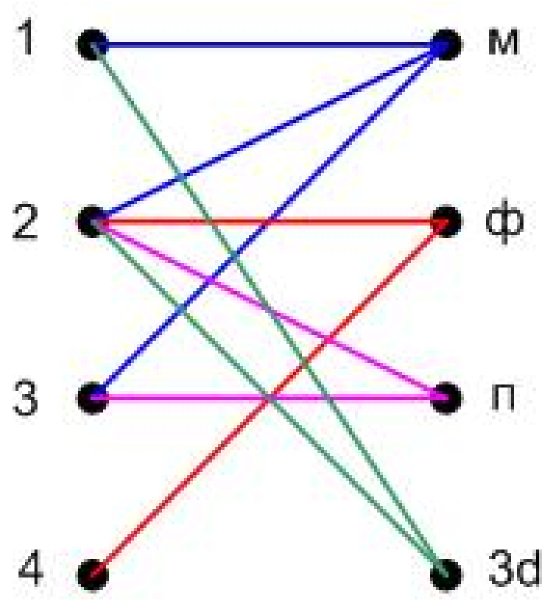


Рис. I.2.6: Граф вариантов расписания

Задача сводится к определению ребер, задающих взаимно однозначное соответствие между множеством дней и множеством мастер-классов.

Из рисунка можно увидеть, что всего количество таких соответствий $k = 3$

Приведем варианты расписания. Из графа видно, что в четверг может быть только физика. Все могут провести мастер-класс во вторник, а в понедельник и в среду согласны поработать вдое специалистов. Если математическое моделирование поставить в понедельник, то во вторник можно провести только 3D моделирование, в противном случае в среду мастер-класс поставить невозможно.

Ответ: 1234, 2134, 2314.

Задача I.2.3.10. (10 баллов)

Миша любит собирать карточки с супергероями.

Всего есть три типа карточек: обычные, редкие и легендарные.

Обычных карточек — 125 видов. Каждого вида было напечатано по 1000 штук.

Редких карточек — 25 видов. Каждого вида было напечатано по 100 штук.

Легендарных карточек — 5 видов. Каждого вида было напечатано по 10 штук.

Карточки запакованы таким образом, что невозможно узнать, какого они вида, пока не откроешь. Мише нравятся 20 видов обычных карточек, 5 видов редких карточек и все виды легендарных. Найдите вероятность того, что, если Михаил купит одну карточку, она окажется именно того вида, который ему нравится?

Решение

Всего карточек $125 \cdot 1000 + 25 \cdot 100 + 5 \cdot 10$.

Всего карточек, которые нравятся Мише: $20 \cdot 1000 + 5 \cdot 100 + 5 \cdot 10$.

Следовательно, вероятность, того, что Миша купит карточку, которая ему нравятся:

$$\frac{20 \cdot 1000 + 5 \cdot 100 + 5 \cdot 10}{125 \cdot 1000 + 25 \cdot 100 + 5 \cdot 10} = \frac{20550}{127550} = 0,1611132889063$$

Ответ: 0,1611132889063.

Вторая попытка. Задачи 10-11 класса

Задача I.2.4.1. (10 баллов)

IT-специалист в первую неделю отпуска израсходовал менее, чем $3/5$ количества взятых с собой денег, во вторую неделю – $1/4$ остатка и еще 3000 руб., в третью неделю – $2/5$ нового остатка и еще 1200 руб. После чего осталось $6/35$ от количества взятых денег. Известно также, что количество денег, оставшихся неизрасходованными к концу первой, второй и третьей недели убывало в арифметической прогрессии.

Сколько денег было израсходовано в три недели отпуска?

Ответ укажите в рублях.

Решение

Обозначим за x общее количество взятых с собой денег, за y – остаток первой недели, а за d – разность арифметической прогрессии. Заполним таблицу:

неделя	израсходовано	остаток	арифметическая прогрессия
1 нед	$< \frac{3}{5}x$	$y > \frac{2}{5}x$	$\frac{6}{35}x + 2d$
2 нед	$\frac{1}{4}y + 30$	$\frac{3}{4}y - 30$	$\frac{6}{35}x + d$
3 нед	$\frac{2}{5}(\frac{3}{4}y - 30) + 12$	$\frac{3}{4}y - 30 - \frac{3}{10}y$	$\frac{6}{35}x$

Приравняем второй и третий столбцы таблицы, и получим систему уравнений:

$$\begin{cases} y = \frac{6}{35}x + 2d, \\ \frac{3}{4}y - 30 = \frac{6}{35}x + d, \\ \frac{3}{4}y - 30 - \frac{3}{10}y = \frac{6}{35}x. \end{cases}$$

Решив систему, получаем $d = 180, y = 600, x = 1400$. Таким образом, за отпуск потрачено $x - \frac{6}{35}x = 1160$ тыс. рублей.

Ответ: 116000 руб.

Задача I.2.4.2. (10 баллов)

В некотором городе стоит дворец техники. На этажах со 2 по 8 включительно располагаются экспонаты выставок. Посетители заходят во дворец на 1 этаже, сдают одежду в гардероб и проходят в холл к лифтам.

В один из моментов на 1 этаже в лифт зашло 7 посетителей. Они поехали наверх к экспонатам. Какова вероятность, что хотя бы на одном уровне вышли по крайней мере два человека?

Решение

Рассмотрим обратную ситуацию: на каждом уровне вышло по одному человеку. Вероятность этого события $P(\bar{A}) = \frac{7!}{7^7} = 0,00612$. Следовательно, вероятность обратного события $P(A) = 1 - 0,00612 = 0,99388$.

Ответ: 0,994. (0,99388).

Задача I.2.4.3. (10 баллов)

Решите уравнение: $\cos\sqrt{x} + \sqrt{\cos x} = 2$.

Решение

Так как $|\cos\sqrt{x}| \leqslant 1$ и $|\sqrt{\cos x}| \leqslant 1$, то уравнение $\cos\sqrt{x} + \sqrt{\cos x} = 2$ равносильно системе уравнений:

$$\begin{cases} \cos\sqrt{x} = 1, \\ \sqrt{\cos x} = 1. \end{cases}$$

Из певрого уравнения системы получим, что $x = 2\pi k$, где $k \in Z$, а из второго уравнения получим, что $\sqrt{x} = 2\pi n \Leftrightarrow x = 4\pi^2 n^2$, где $n \in Z$, $n \geqslant 0$. Найдем пересечение полученных множеств: $2\pi k = 4\pi^2 n^2 \Leftrightarrow k = 2\pi n^2$. Так как π - иррациональное число, то последнее равенство выполяется тогда, и только тогда, когда $n = k = 0$. Следовательно, единственным решением данного уравнения является $x = 0$.

Ответ: $x = 0$.

Задача I.2.4.4. (10 баллов)

Две рамки квадратная и в форме равностороннего треугольника заполнены одинаковым количеством одинаковых шаров, касающихся друг друга и сторон рамок (рамки заполнены, как в бильярде).

Сколько шаров для этого потребуется, если к стороне треугольника примыкает на 14 шаров больше, чем к стороне квадрата?

Решение

Пусть a – количество шаров вдоль стороны квадрата, тогда в сосуде с квадратным дном a^2 шаров. И пусть n – количество шаров вдоль стороны треугольника, тогда в сосуде с треугольным дном будет $1+2+...+n$ шаров. По условию количество шаров в обоих сосудах одинаково, то есть $a^2 = 1+2+...+n$. Сумма арифметической прогрессии в правой части равна $\frac{(1+n)n}{2}$. В левой части воспользуемся условием, что $a = n - 14$. Получим уравнение: $(n-14)^2 = \frac{(1+n)n}{2}$. Раскрыв скобки и приведя подобные, получим квадратное уравнение $n^2 - 57n + 392 = 0$. Оно имеет два корня: $n = 8$ и $n = 49$, первый из которых не подходит по условию $a = n - 14 > 0$. Для второго корня $a = n - 14 = 35$, следовательно всего шаров в каждом из сосудов $35^2 = 1225$.

Ответ: 2450.

Задача I.2.4.5. (10 баллов)

Какую геометрическую фигуру представляют собой множество точек комплексной плоскости, удовлетворяющих условию: $|z| = 1$?

Ответом служит одно слово в именительном падеже.

Решение

Модуль комплексного числа $z = x + yi$ равен $|z| = \sqrt{x^2 + y^2}$. По условию он равен 1: $\sqrt{x^2 + y^2} = 1$, следовательно, $x^2 + y^2 = 1$. Это окружность в комплексной плоскости радиуса 1 с центром в начале координат.

Ответ: Окружность.

Задача I.2.4.6. (10 баллов)

Двое друзей, каждый со своей позиции, ведут наблюдение через вертикальную щель в круглую комнату.

Определить величину щели, если вместе они контролируют только четвертую часть стены комнаты, и при этом угол зрения одного и второго равны соответственно °10 и °20 градусов. При решении считать, что каждый из них видит свой участок стены.

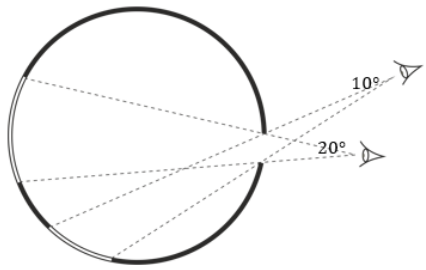


Рис. I.2.7: Схема задачи

Решение

Дополним схему задачи следующими обозначениями, как показано на рисунке. Общую дугу окружности - щель обозначим символом m, а дуги, показывающие сектора обзора, обозначим через n_1 и n_2. Углы обозначающие сектора обзора назовём $\angle A$ и $\angle B$ соответственно.

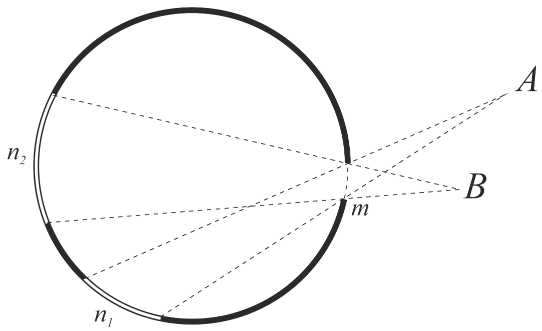


Рис. I.2.8: Схема задачи

Известно, что угол $\angle A$ между секущими, пересекающимися вне окружности, вычисляется по формуле: $\angle A = \frac{n_1 - m}{2}$. Аналогично, $\angle B = \frac{n_2 - m}{2}$. Также по условию задачи $n_1 + n_2 = 90°$. Тогда, складывая левые и правые части двух первых уравнений, получим выражение $\angle A + \angle B = \frac{n_1 + n_2}{2} - m$. Откуда можем найти искомую дугу $m = \frac{n_1 + n_2}{2} - \angle A - \angle B = \frac{90°}{2} - 10° - 20° = 15°$.

Ответ: $^{\circ}15$.

Задача I.2.4.7. (10 баллов)

Каркасный бассейн в форме цилиндра высотой 1.4 метра и внутренним диаметром 3 метра стоит в помещении площадью 24 квадратных метра. Рассчитайте высоту порога при входе в помещение (в см), который необходимо соорудить строителям, чтобы в случае порыва стенок бассейна вода не перетекла через него в смежные помещения.

При расчетах учитывайте, что бассейн наполняется водой до уровня 20 см от верхней кромки.

Решение

Первым шагом рассчитаем объем воды, который находится в бассейне.

$V = \pi R^2(h - \triangle h) = \pi(\frac{3}{2})^2(1.4 - 0.2) \approx 8.48$ куб. метров

Чтобы найти уровень воды L, который получится в помещении в случае порыва бассейна, необходимо разделить полученный объем на площадь помещения $L = V/S \approx 8.48/24 \approx 0.35$ метра, т.е. строителям необходимо оборудовать порог не менее 35 см высотой.

Ответ: 35,3429173529.

Задача I.2.4.8. (10 баллов)

Студия дизайна решила сделать оригинальный сувенир – вазу в виде шара с отсеченным сверху шаровым сегментом. Внутри полость вазы имеет форму перевернутого вверх основанием прямого конуса объемом 1 литр.

Найдите диаметр горлышка вазы, если по задумке дизайнеров объем полости вазы должен был быть максимально возможным.

Введите ответ в дециметрах.

Решение

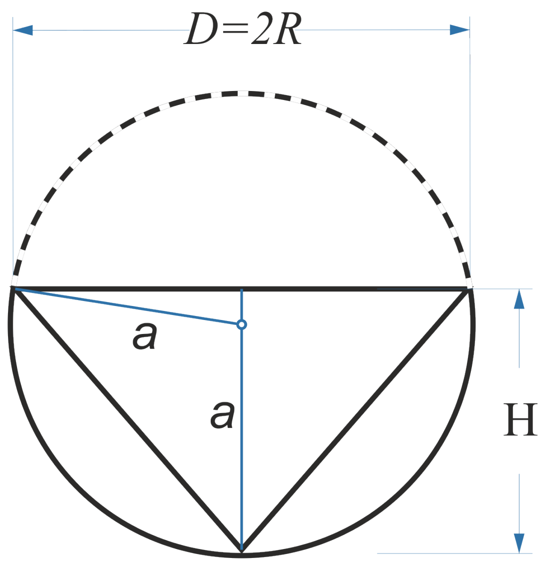


Рис. I.2.9: Проекция вазы на осевую плоскость

Обозначим радиус сферы буквой a, а радиус основания и высоту конуса R и H соответственно. Из рисунка найдём, что радиус горлышка зависит от высоты конуса и радиуса шара. Эта зависимость выражается уравнением $R^2 = a^2 - (H - a)^2$. Тогда выражение для объема конуса в зависимости от высоты и радиуса шара будет иметь вид: $V = \frac{1}{3}\pi(a^2 - (H - a)^2)H$.

Из условия задачи следует, что объем конуса выбран максимальным из тех, которые могут быть вписаны в шар заданного радиуса. Условием максимума объема будет равенство нулю производной от объема по высоте конуса:

$$V(H)^{'} = (\frac{1}{3}\pi(a^2 - (H-a)^2)H)^{'} = 0$$

$$a^2 - (H-a)^2 - 2H(H-a) = 0$$

Откуда получим соотношение между высотой конуса и радиусом шара $H = \frac{4}{3}a$. Подставим полученные соотношения для высоты и радиуса основания конуса в формулу для объема и найдем радиус шара $a = \frac{81}{32}\sqrt[3]{\frac{V}{\pi}}$.

Для диаметра горлышка тогда имеем следующее выражение

$$D = 2R = \frac{4\sqrt{2}}{3}a = \frac{27\sqrt{2}}{8}\sqrt[3]{\frac{V}{\pi}} \approx 32.6$$

Ответ: 1,75461615531.

Задача I.2.4.9. (10 баллов)

Составляя расписание мастер-классов для финала олимпиалы НТИ по профилю "Разработка приложений виртуальной и дополненной реальности"организаторы столкнулись с затруднением. Мастер-классы должны проводится 1 раз в день с понедельника по четверг, но приглашенные специалисты имеют ряд требований к дню проведения мастер-класса. Предложите варианты расписания мастер-классов, если:

а) специалист по математическому моделированию может провести провести мастер-класс в понедельник, вторник или среду;

б) специалист по 3D моделированию может пообщаться с финалистами или в понедельник, или во вторник;

в) мастер-класс по физике может состояться во вторник или четверг;

г) мастер-класс по программированию интерактивного окружения - только во вторник или среду.

Сколько возможно вариантов составления расписания мастер-классов для проведения финала? Укажите их, перечисляя через точку с запятой мастер-классы в порядке их следования в течение финала. Например: Вариант 1. Математическое моделирование; 3D моделирование; программирование интерактивного окружения; физика.

Решение

Построим граф, вершины которого - возможные дни недели (1, 2, 3, 4) и мастер-классы (м, ф, п, 3d). Ребрами соединим мастер-классы с возможными днями проведения(рис. I.2.10).

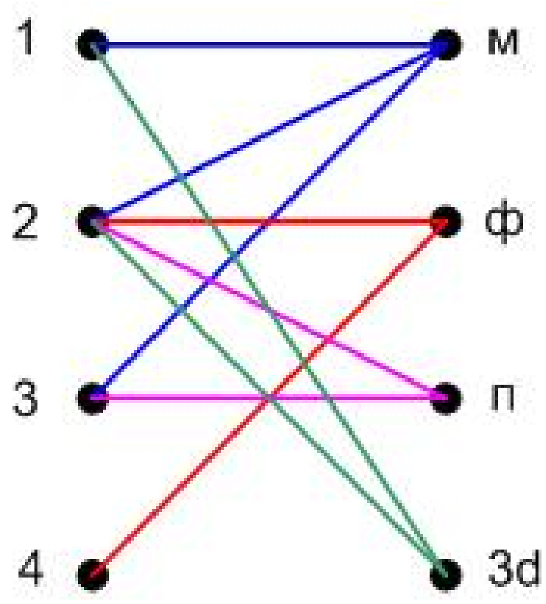


Рис. I.2.10: Граф вариантов расписания

Задача сводится к определению ребер, задающих взаимно однозначное соответствие между множеством дней и множеством мастер-классов.

Из рисунка можно увидеть, что всего количество таких соответствий $k = 3$

Приведем варианты расписания. Из графа видно, что в четверг может быть только физика. Все могут провести мастер-класс во вторник, а в понедельник и в среду согласны поработать вдое специалистов. Если математическое моделирование поставить в понедельник, то во вторник можно провести только 3D моделирование, в противном случае в среду мастер-класс поставить невозможно.

Ответ: 1234, 2134, 2314.

Задача I.2.4.10. (10 баллов)

При каких значениях параметра a функция

$g(x) = \frac{1}{3}x^3 + (a+2)x^2 + (a^2 + 4a - 12)x - 24$ имеет экстремальные точки, принадлежащие промежутку $[-2, 9]$?

Найдите минимальное допустимое значение параметра a, при котором a соответствует условию.

Решение

$g'(x) = x^2 + 2(a+2)x + a^2 + 4a - 12.$

Находим критические точки: $g'(x) = 0$, $x^2 + 2(a+2)x + a^2 + 4a - 12 = 0$

$D_1 = (a+2)^2 - (a^2 + 4a - 12) = 16$, $\sqrt{D_1} = 4.$

$x_1 = -a - 6$; $x_2 = -a + 2$; x_1, x_2- точки экстремума.

По условию

$$\begin{cases} -a-6 \geqslant -2, \\ -a+2 \leqslant 9; \end{cases}$$

$$\begin{cases} a \leqslant -4, \\ a \geqslant -7. \end{cases}$$

Таким образом, $-7 \leqslant a \leqslant -4$.

Ответ: -7.

Второй отборочный этап

Командная часть

Задача II.1.0.1. (100 баллов)

Можно ли узнать возраст клиента на основе информации о его расходах по карте? Мы подготовили задачу на базе реальных банковских транзакций. Совершенствуя свои продукты, банк использует информацию о пользователях, в том числе и возраст. Это помогает сделать персонализированные продукты, которые удовлетворяют реальные потребности клиентов. Но всегда ли календарный возраст соответствует образу жизни (и покупок) человека?

Ваша задача — по информации о расходах клиента банка предсказать, в какую из возрастных групп он попадает. Даны обучающие train данные для построения признаков и обучения моделей, и тестовые test данные для проверки алгоритмов. Это специальным образом подготовленная и анонимизированная информация, на которой можно обучать модели, сохраняя полную безопасность реальных данных клиентов. Решением задачи являются предсказания алгоритмов на тестовых данных. Чем меньше ошибок совершит ваш алгоритм, тем выше вы поднимитесь в рейтинге!

Данные

Для решения задачи участникам предоставляется информация о транзакциях клиентов банка. Объемом около 27000000 миллионов записей. Каждая запись описывает одну банковскую транзакцию. Для каждого из ≈ 20000 тестовых id, участникам необходимо с помощью обученной модели предсказать, в какую из возрастных групп попадает Клиент.

Мы подготовили два набора данных:

1. Обучающий transactions_train.csv, в котором для каждой транзакции известна дата, сумма, тип и id клиента;
2. Тестовый transactions_test.csv, содержащий те же поля:
    - client_id – уникальный номер клиента;
    - trans_date – дата транзакции (представляет из себя просто номер дня в хронологическом порядке, начиная от заданной даты);
    - small_group – группа транзакций, характеризующих тип транзакции (например, продуктовые магазины, одежда, заправки, детские товары и т.п.);
    - amount_rur – сумма транзакции (для анонимизации данные суммы были трансформированы без потери структуры).

На базе данных файлов можно строить различные признаки, которые характеризуют возрастные группы.

Целевая переменная для обучающего датасета находится в файле train_target.csv.

В нем содержится информация о Клиенте и метка возрастной группы, к которой он относится:

- client_id – уникальный номер Клиента (соответствует client_id и файла transactions_train.csv);
- bins – метка возраста. В файле test.csv вам надо предсказать для указанных client_id соответствующую метку группы возраста.

Участникам также предоставлен информационный файл small_group_description.csv, который содержит расшифровку типов транзакций.

Наборы данных: https://onti-2019.s3-eu-central-1.amazonaws.com/public/data_v1.zip

Формат решений

Для каждого примера из тестового набора необходимо предсказать возрастную группу, к которой относится клиент. В систему необходимо предоставить для проверки CSV-файл с предсказаниями, он должен содержать две колонки:

- client_id — идентификатор клиента;
- bins — возрастная группа.

Пример выходных данных:
client_id, bins
0,0
7,1
9,0
10,2
11,1
15,3
...

Задача представляет из себя мультиклассовую классификацию (4 класса – от 0 до 3). Качество решения считается как доля верно угаданных меток возраста по всем тестовым примерам - accuracy.

Для решения удобнее всего использовать язык программирования Python, так как для него есть большое число библиотек для анализа данных: NumPy, Pandas, SciKit-Learn и другие, в качестве инструмента разработки — интерактивную среду Jupyter.

Решение

Импортируем нужные библиотеки.

```
import pandas as pd
import xgboost as xgb
```

Распакуйте архив с данными в папку,где находится этот jupyter notebook (baseline.ipynb). У вас будет папка data, содержащая необходимые файлы.

В данном соревновании перед вами ставится задача предсказания категории возраста, к которой принадлежит клиент банка, на основании его транзакций. В обучающем наборе содержатся информация по транзакциям 30000 клиентов, она находится в файле **transactions_train.csv**. Правильная категория возраста для обучающего набора находится в файле **train_target.csv**.

Считаем данные по транзакциям и правильные ответы.

```
transactions_train=pd.read_csv('data/transactions_train.csv')
train_target=pd.read_csv('data/train_target.csv')
```

Посмотрим на данные.

```
transactions_train.head()
```

	client_id	trans_date	small_group	amount_rur
0	33172	6	4	71.463
1	33172	6	35	45.017
2	33172	8	11	13.887
3	33172	9	11	15.983
4	33172	10	11	21.341

- client_id - уникальный идентификатор клиента
- trans_date - дата совершения транзакции
- small_group - категория покупки
- amount_rur - сумма транзакции

```
train_target.head(5)
```

	client_id	bins
0	24662	2
1	1046	0
2	34089	2
3	34848	1
4	47076	3

- client_id - уникальный идентификатор клиента, соответствует полю client_id из транзакций
- bins - целевая переменная, которую нужно предсказать, это категория возраста клиента

Далее представлен простой вариант решения задачи. Вы можете решать соревнование используя совершенно другой подход.

Посчитаем по каждому клиенту самые простые аггрегационные признаки.

```
agg_features=transactions_train.groupby('client_id')['amount_rur'].agg(['sum', 'mean', \
    'std', 'min', 'max']).reset_index()
agg_features.head()
```

	client_id	sum	mean	std	min	max
0	4	28404.121	39.450168	73.511624	0.043	1341.802

1	6	15720.739	21.535259	26.200397	0.045	315.781
2	7	53630.036	69.379089	253.261383	0.043	4505.971
3	10	34419.365	48.752642	63.191701	0.045	654.893
4	11	26789.404	32.991877	107.395139	0.388	2105.058

Посчитаем для каждого клиента количество транзакций по каждой категории.

```
counter_df_train=transactions_train.groupby(['client_id', \
    'small_group'])['amount_rur'].count()
cat_counts_train=counter_df_train.reset_index().pivot(index='client_id', \
    columns='small_group',values='amount_rur')
cat_counts_train=cat_counts_train.fillna(0)
cat_counts_train.columns=['small_group_'+str(i) for i in cat_counts_train.columns]
cat_counts_train.head()
```

	small_group_0	**small_group_1**	**small_group_2**	...	**small_group_202**
client_id					
0	0.0	226.0	1.0	...	0.0
1	30.0	326.0	0.0	...	0.0
2	21.0	242.0	1.0	...	0.0
3	0.0	156.0	83.0	...	0.0
5	16.0	398.0	1.0	...	0.0
8	29.0	296.0	9.0	...	0.0
9	35.0	222.0	110.0	...	0.0
15	0.0	398.0	0.0	...	0.0
16	0.0	288.0	2.0	...	0.0
21	0.0	185.0	25.0	...	0.0

```
test=pd.merge(test_id,agg_features_test,on='client_id')
test=pd.merge(test,cat_counts_test.reset_index(),on='client_id')
```

В тесте не было некоторых категорий трат, поэтому для того, чтобы обучить модель, нам нужно объединить пространство признаков и train и test.

```
common_features=list(set(train.columns).intersection(set(test.columns)))

y_train=train['bins']
X_train=train[common_features]
X_test=test[common_features]
```

Обучим xgboost на текущих признаках.

```
param={'objective':'multi:softprob','num_class':4,'n_jobs':4,'seed':42}

%%time
model=xgb.XGBClassifier(**param,n_estimators=300)
model.fit(X_train,y_train)
```

CPU times: user 9min 9s, sys: 2.8 s, total: 9min 12s Wall time: 2min 33s

Сделаем предсказание.

```
pred=model.predict(X_test)

pred
```

array([0, 2, 3, ..., 2, 2, 3])

На public лидерборде такое предсказание должно дать качество 0.6118.

Подготовим файл для отправки в систему

```
submission = pd.DataFrame({'bins': pred}, index=test.client_id)
submission.head()
```

client_id	bins
28571	0
27046	2
13240	3
19974	0
10505	1

Сохраняем прогноз на диск в папку submissions. Имя прогноза соответсвует дате и времени его создания, закодированными с помощью timestamp.

```
import time
import os

current_timestamp = int(time.time())
submission_path = 'submissions/{}.csv'.format(current_timestamp)

if not os.path.exists('submissions'):
    os.makedirs('submissions')

print(submission_path)
submission.to_csv(submission_path, index=True)
```

submissions/1573837933.csv

Теперь все готово! Можно отправлять решение.

Заключительный этап

Индивидульный предметный тур

Информатика. 8-11 класс

Задача III.1.1.1. Буквенная нейросеть (25 баллов)

Рассмотрим модель нейросети, состоящей из n слоев, в каждом из которых по 26 нейронов, обозначенных буквами **a, b, ..., z**. Связи между двумя рядом расположенными слоями всегда одинаковые и задаются множеством пар букв. Это значит, например, что если задана связь **ab**, то она есть между буквой **a** первого слоя и буквой **b** второго слоя, между буквой **a** второго слоя и буквой **b** третьего, и так далее. Прохождение по этой сети сигнала можно описать строкой из последовательно соединенных связями букв. Рассмотрим всевозможные способы достижения сигналом слоя номер n. У каждого из этих способов наибольший интерес вызывает последняя буква, образующая выходной сигнал. Требуется по описанию пар связанных букв и числу n расположить буквы n-го слоя в порядке убывания вероятности того, что данная буква образует выходной сигнал. Предполагаем, что сигнал пойдет по любому разрешенному пути с одинаковой вероятностью. Если у двух букв вероятность одинаковая, требуется расположить их в алфавитном порядке.

Формат входных данных

Первая строка содержит два целых числа n и m через пробел – количество слоев сети и количество типов связей между буквами двух соседних слоев соответственно ($1 \leqslant n \leqslant 10$, $1 \leqslant m \leqslant 26^2$). Далее в m строках перечислены пары букв, между которыми имеются соединения. Каждая пара встречается в списке не более одного раза.

Формат выходных данных

Выведите без пробелов строку, состоящую из букв **a, b, ..., z**, упорядоченных необходимым образом.

Пояснения к примеру

Рассмотрим все способы прохождения сигнала до третьего слоя: **abb, aba, abz, acb, acd, bbb, bba, bbz, bab, bac, bzd, cbb, cba, cbz, cde, ded, dec, ede, ecb, ecd, zde**. Среди них 6 оканчиваются на **b**, 4 оканчиваются на **d**, 3 оканчиваются на **a**, 3 оканчиваются на **e**, 3 оканчиваются на **z**, 2 оканчиваются на **c** и для всех остальных букв - по 0 раз. Таким образом, наиболее вероятно, что выходной сигнал образует буква **b**, следующей по вероятности будет буква **d** и так далее...

Примеры

Пример №1

Стандартный ввод
3 11 ab ac bb ba bz cb cd de ed ec zd

Стандартный вывод
bdaezcfghijklmnopqrstuvwxy

Пример программы-решения

Ниже представлено решение на языке C++

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

#define pb push_back
#define sz(a) (int)a.size()
#define all(a) a.begin(), a.end()

using namespace std;
typedef long long ll;

int n, m;
string s;
vector<vector<char> > G('z'+1);
vector<vector<ll> > dp;

bool cmp(char a, char b){
    return (dp[a][n-1] > dp[b][n-1] || (dp[a][n-1] == dp[b][n-1] && a < b));
}

int main(){
    ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n >> m;
    dp.resize('z'+1, vector<ll>(n, 0));
    for(int i = 0; i < m; i++){
        cin >> s;
        G[s[0]].pb(s[1]);
    }


    for(char c = 'a'; c <= 'z'; c++) dp[c][0] = 1;

    for(int i = 0; i < n-1; i++)
        for(char c = 'a'; c <= 'z'; c++)
```

```
            for(auto q : G[c]) dp[q][i+1] += dp[c][i];

    s = "abcdefghijklmnopqrstuvwxyz";
    sort(all(s), cmp);
    cout<<s;
}
```

Задача III.1.1.2. Буквенная нейросеть (35 баллов)

Данная задача отличается от предыдущего формулировкой ограничений на входные данные.

Рассмотрим модель нейросети, состоящей из n слоев, в каждом из которых по 26 нейронов, обозначенных буквами **a, b, . . . , z**. Связи между двумя рядом расположенными слоями всегда одинаковые и задаются множеством пар букв. Это значит, например, что если задана связь **ab**, то она есть между буквой **a** первого слоя и буквой **b** второго слоя, между буквой **a** второго слоя и буквой **b** третьего, и так далее. Прохождение по этой сети сигнала можно описать строкой из последовательно соединенных связями букв. Рассмотрим всевозможные способы достижения сигналом слоя номер n. У каждого из этих способов наибольший интерес вызывает последняя буква, образующая выходной сигнал. Требуется по описанию пар связанных букв и числу n расположить буквы n-го слоя в порядке убывания вероятности того, что данная буква образует выходной сигнал. Предполагаем, что сигнал пойдет по любому разрешенному пути с одинаковой вероятностью. Если у двух букв вероятность одинаковая, требуется расположить их в алфавитном порядке.

Формат входных данных

Первая строка содержит два целых числа n и m через пробел — количество слоев сети и количество типов связей между буквами двух соседних слоев соответственно ($1 \leqslant n \leqslant 100$, $1 \leqslant m \leqslant 26^2$). Далее в m строках перечислены пары букв, между которыми имеются соединения. Каждая пара встречается в списке не более одного раза.

Формат выходных данных

Рассмотрим все способы прохождения сигнала до третьего слоя: **abb, aba, abz, acb, acd, bbb, bba, bbz, bab, bac, bzd, cbb, cba, cbz, cde, ded, dec, ede, ecb, ecd, zde**. Среди них 6 оканчиваются на **b**, 4 оканчиваются на **d**, 3 оканчиваются на **a**, 3 оканчиваются на **e**, 3 оканчиваются на **z**, 2 оканчиваются на **c** и для всех остальных букв - по 0 раз. Таким образом, наиболее вероятно, что выходной сигнал образует буква **b**, следующей по вероятности будет буква **d** и так далее...

Формат выходных данных

Выведите без пробелов строку, состоящую из букв **a, b, . . . , z**, упорядоченных необходимым образом.

Пояснения к примеру

Рассмотрим все способы прохождения сигнала до третьего слоя: **abb, aba, abz, acb, acd, bbb, bba, bbz, bab, bac, bzd, cbb, cba, cbz, cde, ded, dec, ede, ecb, ecd, zde**. Среди них 6 оканчиваются на **b**, 4 оканчиваются на **d**, 3 оканчиваются на **a**, 3 оканчиваются на **e**, 3 оканчиваются на **z**, 2 оканчиваются на **c** и для всех остальных букв - по 0 раз. Таким образом, наиболее вероятно, что выходной сигнал образует буква **b**, следующей по вероятности будет буква **d** и так далее...

Примеры

Пример №1

Стандартный ввод
3 11 ab ac bb ba bz cb cd de ed ec zd

Стандартный вывод
bdaezcfghijklmnopqrstuvwxy

Пример программы-решения

Ниже представлено решение на языке C++

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

#define pb push_back
#define sz(a) (int)a.size()
#define all(a) a.begin(), a.end()

using namespace std;
typedef long long ll;
//-----------------------------------------------------
typedef vector<int> lnum;
const int base = 1000*1000*1000;
int n, m;
string s;
vector<vector<char>> G('z'+1);
vector<vector<lnum>> dp;

inline lnum sum(lnum a, lnum b){
    int carry = 0;
    for (size_t i=0; i<max(a.size(),b.size()) || carry; ++i) {
        if (i == a.size())
            a.push_back (0);
        a[i] += carry + (i < b.size() ? b[i] : 0);
```

```
        carry = a[i] >= base;
        if (carry)  a[i] -= base;
    }
    return a;
}

void print(lnum a){
    printf ("%d", a.empty() ? 0 : a.back());
    for (int i=(int)a.size()-2; i>=0; --i)
        printf ("%09d", a[i]);
}
//--------------------------------------

bool cmp(char c1, char c2){
    lnum a = dp[c1][n-1];
    lnum b = dp[c2][n-1];

    if(sz(a) > sz(b)) return 1;
    if(sz(a) == sz(b)){
        for(int i = sz(a)-1; i >= 0; i--){
                if(a[i] > b[i]) return 1;
                if(a[i] < b[i]) return 0;
        }
    }
    if(c1 < c2) return 1;
    return 0;
}

int main(){
    ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n >> m;
    dp.resize('z'+1, vector<lnum>(n));
    for(int i = 0; i < 'z'+1; i++) for(int j = 0; j < n; j++) dp[i][j].pb(0);
    for(int i = 0; i < m; i++){
        cin >> s;
        G[s[0]].pb(s[1]);
    }

    for(char c = 'a'; c <= 'z'; c++) {
            dp[c][0].resize(1);
            dp[c][0][0] = 1;
    }

    for(int i = 0; i < n-1; i++)
        for(char c = 'a'; c <= 'z'; c++)
            for(auto q : G[c]) dp[q][i+1] = sum(dp[c][i], dp[q][i+1]);

    s = "abcdefghijklmnopqrstuvwxyz";
    sort(all(s), cmp);
    cout<<s;
}
```

Задача III.1.1.3. D-разделимость (20 баллов)

В данной задаче нас будут интересовать только количественные характеристики каждого слоя некой нейросети. Пусть в этой сети имеется n слоев, i-й слой состоит из a_i нейронов. Кроме того, задано неотрицательное пороговое значение D. Будем говорить, что два слоя j-й и k-й $(j < k)$ D-разделены, если суммарно во всех слоях между ними не менее D нейронов. Другими словами, если верно неравенство

$$\sum_{i=j+1}^{k-1} a_i \geqslant D.$$

В частности, любые два слоя 0-разделены.

По заданным размерам каждого слоя и пороговому значению D требуется найти, сколько в этой сети пар D-разделенных слоев.

Формат входных данных

Первая строка содержит два целых числа n и D через пробел — количество слоев сети и пороговое значение $(1 \leqslant n \leqslant 10^5, \quad 1 \leqslant D \leqslant 10^9)$. Во второй строке содержится n чисел a_i через пробел $(1 \leqslant a_i \leqslant 10^9)$.

Формат выходных данных

Выведите одно целое число - количество пар D-разделенных слоев.

Пояснения к ответу

В примере есть 16 5-разделенных пар слоев: 1-3, 1-4, 1-5, 1-6, 1-7, 1-8, 2-5, 2-6, 2-7, 2-8, 3-5, 3-6, 3-7, 3-8, 4-8, 5-8.

Примеры

Пример №1

Стандартный ввод
8 5 3 7 2 5 1 1 4 2

Стандартный вывод
16

Пример программы-решения

Ниже представлено решение на языке C++

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

#define pb push_back
#define sz(a) (int)a.size()
#define all(a) a.begin(), a.end()

using namespace std;
typedef long long ll;

ll n, D;

```

```cpp

int main(){
    ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n >> D;

    vector<ll> v(n);
    for(int i = 0; i < n; i++) cin >> v[i];

    ll sum = 0, l = 0, r = 0, ans = 0;
    bool ok = 0;
    while(!ok){
        if(sum < D) {
            r++;
            if(r-1 < n)sum += v[r-1];
            if(sum >= D) ans += n-r+1;
            if(r > n) ok = 1;
        }
        else
        if(sum >= D){
            l++;
            sum -= v[l-1];
            if(sum >= D) ans += n-r+1;
               if(l >= n) ok = 1;
        }
    }
    cout << ans;
}
```

Задача III.1.1.4. Сеть на плоскости (20 баллов)

А теперь давайте изобразим нейросеть на плоскости. В этой задаче она будет состоять всего из трех слоев. Первый (входной) слой состоит только из одного элемента A и расположен он в точке с координатами $(-1, 0)$. Второй (промежуточный) слой состоит из n элементов B_i, все они имеют координаты $(0, y_i)$. Третий (выходной) слой состоит из m элементов C_j, все они имеют координаты $(1, y_j)$. Между некоторыми элементами второго слоя и элементом A имеются связи, аналогично, между некоторыми элементами третьего слоя и некоторыми элементами второго слоя так же есть связи. Будем считать, что эти связи на плоскости соответствуют отрезкам, соединяющим соответствующие точки. В итоге, получим множество четырехугольников вида $AB_{i_1}C_jB_{i_2}$. Если все координаты являются целыми числами, то площадь любого четырехугольника $AB_{i_1}C_jB_{i_2}$ является целочисленной. Найдите площадь самого большого такого четырехугольника. Если ни одного четырехугольника нет — вывести 0.

Формат входных данных

Первая строка содержит число n. Во второй строке содержится n целых чисел через пробел — координаты y_i элементов второго слоя. Третья строка содержит число m. В четвертой строке содержится m целых чисел через пробел — координаты y_j элементов третьего слоя. Числа во второй строке попарно не совпадают и числа в третьей строке попарно не совпадают. Во второй и четвертой строке числа приведены в порядке возрастания. Далее, в пятой строке содержится число k — количество связей между элементом A и элементами второго слоя. В шестой строке содержится k

целых чисел через пробел — номера элементов второго слоя, с которыми соединяется элемент A. В седьмой строке содержится число s. В следующих s строках содержатся s пар номеров связанных элементов, первое из пары — номер элемента второго слоя, второе — номер элемента третьего слоя. Элементы слоев пронумерованы, начиная с 1, в соответствии с порядком своего появления во входных данных. Все числа во во входных данных по абсолютному значению не превосходят 10^5. Числа n, m, k, s — положительные.

Формат выходных данных

Выведите одно целое число — площадь самого большого получившегося четырехугольника.

Примеры

Пример №1

Стандартный ввод

```
6
-4 -2 -1 2 4 5
7
-5 -4 -2 1 2 3 5
5
6 4 5 3 1
18
6 7
5 7
5 6
4 6
3 6
2 6
3 4
4 5
4 3
5 5
3 3
3 1
2 2
2 1
1 2
1 1
1 3
4 7
```

Стандартный вывод

```
6
```

Пример программы-решения

Ниже представлено решение на языке C++

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

```

```cpp
#define pb push_back
#define sz(a) (int)a.size()
#define all(a) a.begin(), a.end()

using namespace std;
typedef long long ll;

int n, m, k, s;

int main(){
    ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    cin >> n;
    vector<int> b(n+1);
    for(int i = 1; i <= n; i++) cin >> b[i];

    cin >> m;
    vector<int> c(n+1);
    for(int i = 1; i <= m; i++) cin >> c[i];

    cin >> k;
    vector<int> A(1e5+1, 0);
    int d;
    for(int i = 0; i < k; i++){
        cin >> d;
        A[d] = 1;
    }

    cin >> s;
    int l, r;
    vector<vector<int> > G(m+1);

    for(int i = 0; i < s; i++){
        cin >> l >> r;
        if(A[l]) G[r].pb(l);
    }

    for(int i = 1; i <= m; i++) sort(all(G[i]));

    int mx = 0;
    for(int i = 1; i <= m; i++) if(sz(G[i]) > 1) mx = max(mx, b[G[i].back()] -
    ↪ b[G[i][0]]);

    cout<<mx;
}
```

Математика. 8-9 класс

Задача III.1.2.1. (15 баллов)

Какими должны быть значения a и b, чтобы многочлен $x^4 + 3x^3 - x^2 + ax + b$ был полным квадратом?

Решение

Приведённый многочлен четвёртой степени может быть квадратом лишь приведённого квадратного трёхчлена. Следовательно,

$x^4 + 3x^3 - x^2 + ax + b = (x^2 + px + q)^2.$

Возведя в квадрат трёхчлен, стоящий в правой части, и приравняв коэффициенты при одинаковых степенях аргумента в обеих частях тождества, получим

$2p = 3, p^2 + 2q = -1, 2pq = a, q^2 = b.$

Решив эту систему уравнений, найдём $p = 3/2, q = -13/8, a = -39/8, b = 169/64.$

Система оценки

Критерий	**Начисляемые баллы**
Определен вид квадратного двучлена для приведенного многочлена четвертой степени	4
Приведеная система уравнений, которые позволяют вычислить значения искомых параметров	7
Корректно вычислены значения искомых параметров	4

Ответ: $a = -39/8, b = 169/64.$

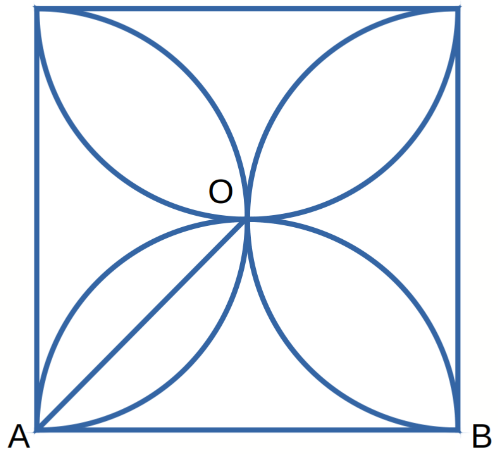
Задача III.1.2.2. (15 баллов)

Для обеспечения безопасности доступа к персональным данным в серверной комнате системный администратор создал специальный плоский электронный ключ-карточку, принцип работы которой заключается в следующем: при совмещении ключа с электронным считывателем дверь открывается, если вид фигуры и его площадь совпадут с данными из списка допущенных в комнату сотрудников.

На работу устроился новый сотрудник, для которого системный администратор изготовил следующим образом ключ: внутри квадрата со стороной 5 см на каждой его стороне как на диаметре построена полуокружность; далее на розетку, ограниченную дугами полуокружностей, наносится специальное отражающее покрытие, площадь которого считывается электронным датчиком и сравнивается со списком допущенных в комнату сотрудников. Какую величину площади (в кв.см) должен внести списки доступа системный администратор для этого сотрудника? Число π принято равным 3.14.

Решение

Хорда OA стягивает дугу в 90^o (см. рисунок); поэтому площадь половины лепестка равна $\frac{\pi a^2}{16} - \frac{a^2}{8} = \frac{a^2(\pi-2)}{16}$.



Следовательно, искомая площадь

$$S = 8 \cdot \frac{a^2(\pi - 2)}{16} = \frac{a^2(\pi - 2)}{2} = 14.25$$

Система оценки

Критерий	Начисляемые баллы
Аргументированно определена площадь одного выделенного элемента розетки (сегмента или сектора)	8
Определена площадь всей розетки	4
Правильно вычислена площадь всей розетки	3

Ответ: 14.25.

Задача III.1.2.3. (20 баллов)

Найти все положительные корни уравнения: $\sqrt{z^2 - z - 1} + \sqrt{z^2 + z + 3} = \sqrt{2z^2 + 8}$

Решение

После возведения уравнения в квадрат получим

$$2z^2 + 2 + \sqrt{(z^2 - z - 1)(z^2 + z + 3)} = 2z^2 + 8,$$

т.е.

$$\sqrt{(z^2 - z - 1)(z^2 + z + 3)} = 3$$

Отсюда

$$(z^2 - z - 1)(z^2 + z + 3) = 9,$$

или

$$z^4 + z^2 - 4z - 12 = 0.$$

Заметим, что одним из корней последнего уравнения является 2. Разделив на $z - 2$, приходим к уравнению $z^3 + 2z^2 + 5z + 6 = 0$, которое не имеет положительных корней, так как все его коэффициенты положительны, следовательно, корень будет единственным.

Система оценки

Критерий	**Начисляемые баллы**
Выполнено возведение уравнения в квадрат	4
Найден один из корней уравнения	5
Аргументировано показано, что полученный корень - единственный	11

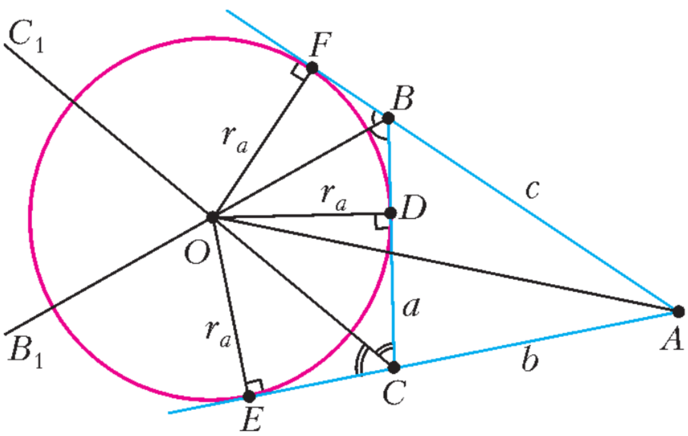
Ответ: 2.

Задача III.1.2.4. (20 баллов)

Найти стороны треугольника, если радиусы вневписанных окружностей равны $3, 2, \frac{7}{6}$.

Решение

Вычислим площадь треугольника ABC через радиус вневписанной окружности.



При этом заметим, что центр вневписанной окружности O находится в точке пересечения биссектрис BB_1 и CC_1 соответствующих внешних углов треугольника

ABC. Соединим точки A и O и рассмотрим площади треугольников:

$$S_{ABO} = \frac{cr_a}{2}, S_{ACO} = \frac{br_a}{2}, S_{BCO} = \frac{ar_a}{2}.$$

Поскольку $S_{ABC} = S = S_{ABO}+S_{ACO}-S_{BCO}$ или $S = \frac{r_a(b+c-a)}{2}$, то, используя выражение для полупериметра треугольника, можно выразить радиус окружности через площадь и соответствующую сторону треугольника: $r_a = \frac{2S}{2p-2a} = \frac{S}{p-a}$. Аналогично определяются и два других радиуса:

$$r_a = \frac{S}{p-a}, r_b = \frac{S}{p-b}, r_c = \frac{S}{p-c} \tag{III.1.1}$$

Из выражений (III.1.1) выразим через радиусы окружностей стороны:

$$\frac{1}{r_a} + \frac{1}{r_b} = \frac{2p-a-b}{S} = \frac{c}{S}, \frac{1}{r_a} + \frac{1}{r_c} = \frac{b}{S}, \frac{1}{r_b} + \frac{1}{r_c} = \frac{a}{S} \tag{III.1.2}$$

Из формул (III.1.2) найдем отношения сторон треугольника:

$$a : b : c = \left(\frac{1}{r_b} + \frac{1}{r_c}\right) : \left(\frac{1}{r_a} + \frac{1}{r_c}\right) : \left(\frac{1}{r_a} + \frac{1}{r_b}\right),$$

откуда

$$a = \left(\frac{r_a r_c + r_a r_b}{r_a r_c + r_b r_c}\right) c, b = \left(\frac{r_a r_b + r_b r_c}{r_a r_c + r_b r_c}\right) c \tag{III.1.3}$$

Из формул (III.1.1) выразим сумму попарных произведений радиусов вневписанных окружностей:

$$r_a r_c + r_b r_c = S^2\left(\frac{1}{(p-a)(p-b)} + \frac{1}{(p-a)(p-c)} + \frac{1}{(p-b)(p-c)}\right) = \frac{S^2 p}{(p-a)(p-b)(p-c)} =$$

$$= \frac{S^2 p^2}{p(p-a)(p-b)(p-c)} = \frac{S^2 p^2}{S^2}$$

Значит, полупериметр треугольника равен

$$p = \sqrt{r_a r_b + r_a r_c + r_b r_c} \tag{III.1.4}$$

Из формул (III.1.3) и (III.1.4) непосредственно следует, что

$$c = \frac{r_c(r_a + r_b)}{\sqrt{r_a r_b + r_a r_c + r_b r_c}}, \ b = \frac{r_b(r_a + r_c)}{\sqrt{r_a r_b + r_a r_c + r_b r_c}}, \ a = \frac{r_a(r_b + r_c)}{\sqrt{r_a r_b + r_a r_c + r_b r_c}} \tag{III.1.5}$$

Подставив данные в условии задачи числовые значения радиусов, вычислим длины сторон:

$a = 1.7, \ b = 2.5, \ c = 2.8$

Система оценки

Критерий	Начисляемые баллы
Аргументированно выражены радиусы окружностей через площадь треугольника и его стороны	3
В преобразовании тригонометрического выражения применена формула понижения степени	4
Две стороны треугольника выражены через третью сторону (выполнен вывод выражения)	5
Корректно записано окончательные выражения для длин сторон треугольника	5
Корректно выполнены численные расчеты	3

Ответ: 1.7, 2.5, 2.8.

Задача III.1.2.5. (30 баллов)

Для шифрования личного электронного дневника пользователь решил использовать числовой пароль. Число он решил выбрать таким, чтобы оно было равно сумме квадратов всех членов некоторой конечной геометрической последовательности (упорядоченного набора из n чисел, где каждое последующее число в одно и то же число раз q больше или меньше, чем текущее). Однако само число записывать пользователь не рискнул, а записал только первый $b_1 = x$ и последний $b_n = y$ члены прогрессии, а также сумму $S_n = S$ всех ее членов. Запишите выражение, с помощью которого пользователь сможет, используя записанные параметры x, y, S быстро получить значение числового пароля.

Решение

Пусть $b_1, b_2, ..., b_n$ - данная конечная геометрическая прогрессия со знаменателем q и суммой n слагаемых, которая равна S_n. Требуется найти

$$b_1^2 + b_2^2 + b_3^2 ... + b_n^2 = b_1^2 + b_1^2q^2 + b_1^2q^4 + ... + b_1^2q^{2n} = \qquad \text{(III.1.6)}$$
$$= b_1^2(1 + q^2 + q^4 + ... + q^{2n}) = b_1\frac{1 - q^{2n}}{1 - q^2}$$

Согласно условию задачи имеем

$$\begin{cases} xq^{n-1} = y \\ \frac{x(1-q^n)}{1-q} = S \end{cases}$$

Из первого уравнения следует, что $q^{n-1} = \frac{y}{x}$, т.е. $q^n = \frac{y}{x}q$. Подставив это выражение во второе уравнение системы, получим

$$\frac{x(1 - \frac{y}{x}q)}{1 - q} = S$$

$$x - yq = S - Sq$$
$$q(S - y) = S - x$$
$$q = \frac{S - x}{S - y}$$

Значит,

$$q = \frac{S - x}{S - y} \tag{III.1.7}$$

Теперь подставим выражение (III.1.7) в равенство (III.1.6)

$$x^2 \left(\frac{1 - \frac{y^2(S-x)^2}{x^2(S-y)^2}}{1 - \frac{(S-x)^2}{(S-y)^2}} \right) = \frac{x^2 - \frac{y^2(S-x)^2}{(S-y)^2}}{1 - \frac{(S-x)^2}{(S-y)^2}} = \frac{x^2(S-y)^2 - y^2(S-x)^2}{(S-y)^2 - (S-x)^2} =$$
$$= \frac{(xS - xy - Sy + xy)(xS - xy + Sy - xy)}{(S - y - S + y)(S - y + S - x)} =$$
$$= \frac{S(x-y)((x+y)S - 2xy)}{(x-y)(2S - (x+y))} = \frac{(x+y)S - 2xy}{(2S - (x+y))} S.$$

Система оценки

Критерий	**Начисляемые баллы**
Записано выражение (III.1.6)	10
Получено выражение (III.1.7)	10
Получено корректное выражение для суммы	10

Ответ: $\frac{(x+y)S-2xy}{(2S-(x+y))}S$

Математика. 10-11 класс

Задача III.1.3.1. (15 баллов)

Хеш-функция f для определения уникальных идентификаторов для получения доступа пользователей к данным некоторого сайта выбрана такой, что для любых входных параметров x и y выполняется равенство $f(x/y) = f(y) - 3f(x) + 14x/y$. Найдите $f(0.25)$, если $f(2) = 3$.

Решение

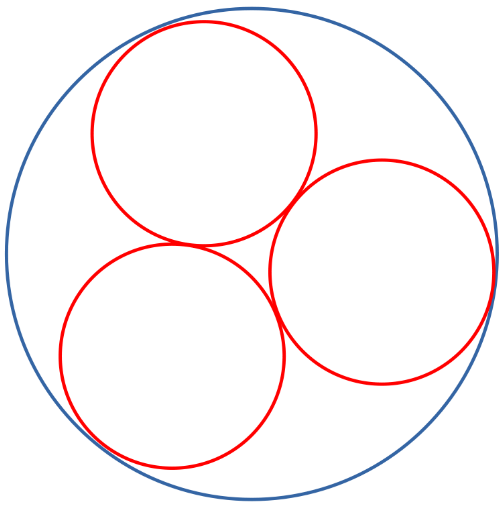
$$f(1) = f(2/2) = f(2) - 3f(2) + 14 \cdot 2/2 = 8$$
$$f(1/2) = f(2) - 3f(1) + 14 \cdot 1/2 = -14$$
$$f(1/4) = f(\tfrac{1/2}{2}) = f(2) - 3f(1/2) + 14\tfrac{1/2}{2} = 58.5$$

Критерий	Начисляемые баллы
Приведен верный ответ без вычислений	5
Приведен ход вычислений, но допущена ошибка	10
Приведен ход вычислений,ошибок в вычислениях нет	15

Ответ: 58.5.

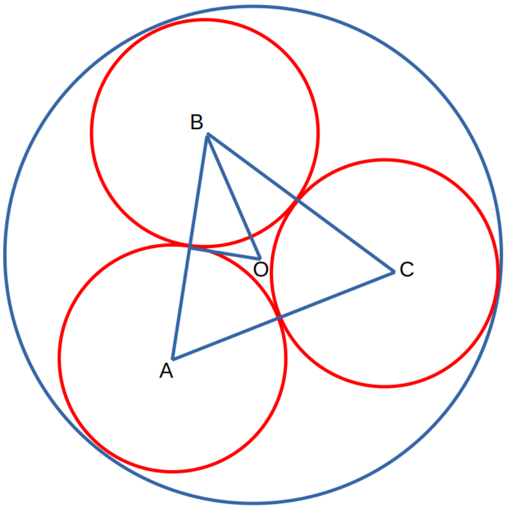
Задача III.1.3.2. (15 баллов)

Рабочее место трех инженеров кибербезопасности представляет из себя круглую платформу, на которой стоят три круглых стола, вид сверху на которые представлен на рисунке. Найдите площадь заглушки для области, ограниченной тремя столами, если радиус платформы R равен 3 метрам. Число π принять равным 3.14.



Решение

Отметим центры окружностей и проведем через эти точки равносторонний треугольник со сторонами, равными диаметру окружностей. Искомая площадь $S = S_{ABC} - S_1$, где S_1 - площадь сектора, ограниченного дугой окружности и половинами сторон треугольника (см. рисунок).



Пусть x - радиус столов. Тогда $S = \frac{(2x)^2\sqrt{3}}{4} - 3 \cdot \frac{\pi x^2}{6} = x^2\sqrt{3} - \frac{\pi x^2}{2} = = \frac{x^2(2\sqrt{3}-\pi)}{2}$. Учитывая, что $AB = 2x = OB\sqrt{3}$ и $OB = R - x$, получаем $x = \frac{R\sqrt{3}}{2+\sqrt{3}} = R(2\sqrt{3} - 3)$. Окончательно

$$S = \frac{R^2(2\sqrt{3}-3)^2(2\sqrt{3}-\pi)}{2} = \frac{3R^2(7-4\sqrt{3})(2\sqrt{3}-\pi)}{2}$$

Поставив значение радиуса, получим, $S = 0.4968$.

Система оценки

Критерий	**Начисляемые баллы**
Корректно выполнены дополнительные построения треугольника	4
Сформулировано, как вычисляется площадь области	2
Представлено правильное выражение для площади	6
Корректно выполнены расчеты	3

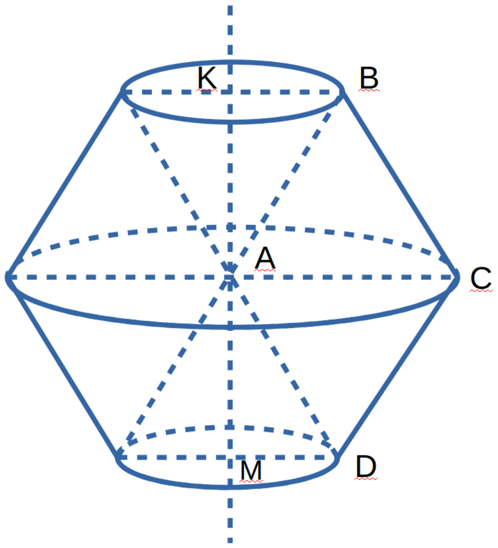
Ответ: 0.4968.

Задача III.1.3.3. (20 баллов)

Согласно инструкции компании N об организации и обеспечения безопасности хранения данных при создании резервной копии информации носитель, на который была записана информация, должен быть помещен в специальный контейнер. Контейнер представляет собой фигуру вращения, полученную следующим образом: параллелограмм со сторонами a и b, вращается вокруг оси, перпендикулярной диагонали длиной d и проходящей через ее конец. Сотрудник решил поместить контейнер в дополнительный пластиковый чехол, причем такой, чтобы он плотно прилегал ко всей поверхности контейнера без образования воздушных пустот. Какова будет площадь поверхности этого чехла?

Решение

Тело вращения представлено на рисунке.



Его поверхность S состоит из боковых поверхностей двух усеченных конусов, полученных при вращении отрезков BC и CD, и двух конусов, полученных при вращении отрезков AB и AD. Пусть $AB = a$, $AD = b$. Таким образом,

$$S = \pi(KB + AC)BC + \pi(MD + AC)CD + \pi KB \cdot AB + \pi MD \cdot AD.$$

Преобразуем это выражение, учитывая, что $AD = BC = b$, $CD = AB = a$, $KB + MD = AC$. Получим

$$S = \pi(KB \cdot BC + AC \cdot BC + MD \cdot CD + AC \cdot CD + \pi KB \cdot AB + \pi MD \cdot AD) =$$

$$= \pi((KB + MD)BC + (KB + MD)AB + AC \cdot BC + AC \cdot AB =)$$
$$= 2\pi(BC + AB)AC$$

Так как по условию $AC = d$, то $S = 2\pi d(a + b)$.

Система оценки

Критерий	Начисляемые баллы
Корректно выполнено построение фигуры вращения	5
Корректно записано исходное выражение для площади	6
Верно выполнено окончательное вычисление площади	9

Ответ: $2\pi d(a + b)$.

Задача III.1.3.4. (20 баллов)

Найти наибольшее значение, которое может принять выражение:

$$a\sin x + b\sin(\frac{\pi}{3} + x) + c\sin(\frac{\pi}{3} - x).$$

Решение

Согласно неравенству Коши-Буняковского

$$a\sin x + b\sin(\frac{\pi}{3} + x) + c\sin(\frac{\pi}{3} - x) \leqslant \sqrt{a^2+b^2+c^2}\cdot\sqrt{\sin^2 x + \sin^2(\frac{\pi}{3}+x) + \sin^2(\frac{\pi}{3}-x)}.$$

Покажем, что $\sin^2 x + \sin^2(\frac{\pi}{3}+x) + \sin^2(\frac{\pi}{3}-x) = \frac{3}{2}$

$$\sin^2 x + \sin^2(\frac{\pi}{3}+x) + \sin^2(\frac{\pi}{3}-x) = \frac{1}{2}(1-\cos 2x) + \frac{1}{2}(1-\cos(\frac{2\pi}{3}+2x)) + \frac{1}{2}(1-\cos(\frac{2\pi}{3}-2x)) =$$

$$= \frac{3}{2} - \frac{1}{2}(\cos 2x + \cos(\frac{2\pi}{3}+2x) + \cos(\frac{2\pi}{3}-2x)) = \frac{3}{2} - \frac{1}{2}(\cos 2x + 2\cos\frac{2\pi}{3}\cos(2x)) = \frac{3}{2}$$

Следовательно,

$$\sqrt{a^2+b^2+c^2}\cdot\sqrt{\sin^2 x + \sin^2(\frac{\pi}{3}+x) + \sin^2(\frac{\pi}{3}-x)} = \sqrt{\frac{3}{2}(a^2+b^2+c^2)}$$

Система оценки

Критерий	**Начисляемые баллы**
Верно применено неравенство Коши-Буняковского	10
В преобразовании тригонометрического выражения применена формула понижения степени	8
В преобразовании тригонометрического выражения применена формула преобразования суммы в произведение	8
Корректно записано окончательное выражение	8

Ответ: $\sqrt{\frac{3}{2}(a^2+b^2+c^2)}$.

Задача III.1.3.5. (30 баллов)

Системному администратору для организации олимпиады поставили задачу соединить 16 компьютеров в кольцо. После этого все компьютеры должны быть попарно соединены восемью оптовололоконными кабелями так, чтобы у них не было общих точек соединения и не пересекались. Сколькими способами системный администратор может это сделать?

Решение

Геометрически задача заключается в соединении $2n$ точек на окружности n непересекающимися хордами.

Пусть a_n – количество способов соединить $2n$ точек на окружности n непересекающимися хордами. Очевидно, что $a_1 = 1$ и $a_2 = 2$. Покажем, что

$$a_n = a_{n-1} + a_{n-2}a_1 + a_{n-3}a_2 + a_1a_{n-2} + a_{n-1} \text{ (числа Каталана).}$$

Фиксируем одну из данных $2n$ точек. Хорда, выходящая из неё, делит окружность на две дуги, причём на каждой дуге расположено чётное число данных точек. Если на одной дуге расположено $2k$ точек, то на другой – $2(n-k-1)$ точек; эти точки можно соединить непересекающимися хордами (не пересекающими первую хорду) $a_{n-k-1}a_k$ способами. Осталось просуммировать по k от 0 до $2n-2$.

$a_3 = a_2 + a_1a_1 + a_2 = 5$;

$a_4 = a_3 + a_2a_1 + a_1a_2 + a_3 = 14$;

и т.д.

Таким образом, $a_8 = 1430$.

Система оценки

Критерий	**Начисляемые баллы**
Доказана рекуррентная формула вычисления a_n с рассуждениями	10
Выведено число a_n для первых 4 - 5 значений	10
Корректно проведены вычисления для окончательного ответа	10

Ответ: 1430.

Командный практический тур

Задача командного тура

В командной части заключительного этапа участникам было предложено решить реальную бизнес-задачу по искусственному интеллекту.

Длительность командного этапа – 2 дня (14,5 астрономических часов).

Участникам олимпиады предстояло работать в командах по два человека. При этом в одной команде могли быть участники из разных возрастных групп. Для участников, которые не сформировали команду до финала можно было сделать это через чат: https://vk.me/join/AJQ1d562AxaWeJreGIY_cGQj, а также с помощью модераторов на сессии по командообразованию, которая состоялась в начале финала.

Командам было предложено 2 набора обезличенных данных о транзакциях 50 000 клиентов ПАО Сбербанк (далее – «Банк») в течение одного года. Один набор предназначался для обучения модели, второй – для проверки. Каждый из них содержал данные о транзакциях 25 000 различных клиентов. Задача участников заключалась в предсказании вероятности совершения покупки клиентом в каждой из 8 целевых категорий в течение ближайшей недели.

Актуальность задачи

Решение предложенной задачи является важным шагом к построению рекомендательной системы для клиентов Банка. Например, на основании таких систем Банк и компании экосистемы Банка в дальнейшем может выстраивать систему акций и специальных предложений для клиентов. Для удобства пользователей, а также достижения бо́льшего эффекта важно, чтобы рекомендации соответствовали предпочтениям конкретных пользователей и были приближены к их ожиданиям.

В рамках задачи команды должны были построить модель, которая определяла бы вероятность совершения покупки в ближайшую неделю в определенных категориях товаров, что соответствует реальной задаче, с которой приходится сталкиваться в работе специалистам. На основе результатов, полученных в ходе обработки данных разработанной моделью, определяется, какие рекомендации показать тому или иному клиенту: чем выше вероятность покупки в данной категории, тем актуальнее информация о ней.

Ход работы

Для решения инженерной задачи участникам предоставлялись ноутбуки с одинаковой комплектацией программного обеспечения и вычислительной мощностью (далее – «Оборудование»).

Описание задачи было представлено на сайте: https://onti.ai-academy.ru/competition

Решения задачи необходимо также загружать на сайт.

Участники могли самостоятельно установить дополнительные программные продукты, которые, по их мнению, необходимы для решения задачи только в случае, если данные программные продукты распространяются свободно. Участники не имели права использовать для решения задачи программное обеспечение, на которое у них есть личная лицензия.

В случае возникновения сбоя в работе оборудования участники могли потребовать увеличение предоставляемого на решение задачи времени в день, когда произошел описываемый сбой, и на время, соответствующее времени, которое прошло между моментом возникновения описываемого сбоя и его устранением.

После размещения участником решения задачи на сайте производился автоматический перерасчет общего рейтинга команд участников, который размещался в открытом доступе на сайте и определялся автоматически на основе значения выбранной метрики, полученной участниками по результатам решения задач.

Качество предложенных решений задач рассчитывалось автоматически на основе выбранной для задачи метрики. Описание метрики находится в разделе «Метрика и формат решений».

После загрузки финального решения участники должны были предоставить итоговый код алгоритма, благодаря которому решение было сформировано, на адрес электронной почты mikheeva@nti-contest.ru не позднее срока, указанного в расписании. Если решение было предоставлено в форме ссылки на размещенный в сети интернет код, решение должно быть доступно в исходном виде до «01» сентября 2020 г.

После формирования рейтинга участников была проведена защита решений задачи для первых 20 участников рейтинга перед комиссией, цель которой удостовериться в самостоятельности полученных решений. При неуспешной защите комиссия оставляла за собой право отказать Участнику в дальнейшем участии в финале в любое время без возмещения каких-либо убытков и расходов.

Участники были обязаны:

- Выполнять задания лично, а также не нарушать авторские и иные интеллектуальные права третьих лиц при подготовке решения задач и разработке алгоритма в рамках финала, в том числе не использовать результаты интеллектуальной деятельности, права на которые принадлежат третьим лицам.
- Работать над решением задачи в строго установленное организатором время согласно установленному расписанию работы.
- Не использовать программное обеспечение, нарушающее работу сайта и автоматизированных систем организаторов, а также создающее возможность изменения результатов соревнования. В случае выявления организаторами попыток выведения из строя системы организаторов, участник, предпринявший такие попытки, мог лишиться прав на участие в соревновании и получение приза.
- Не использовать проприетарное программное обеспечение, требующее приобретения соответствующих лицензий, в решениях задач.
- Обеспечивать конфиденциальность аутентификационных сведений (логин-пароль), необходимых для загрузки решений на сайт.

Описание данных

Обучающий (transactions_train.csv) и тестовый (transactions_test.csv) наборы данных идентичны и содержат информацию по транзакциям 25 000 уникальных клиентов каждый в следующем формате:

- client_dk – уникальный номер клиента;
- trans_date – дата транзакции (представляет из себя номер дня в хронологическом порядке, начиная от заданной даты);
- small_group – группа транзакций, характеризующих тип транзакции (например, продуктовые магазины, одежда, заправки, детские товары и т.п.);
- amount – сумма транзакции в условных единицах.

Для обучающего набора в файле с правильными ответами (train_target.csv) находится таблица размера 25000 x 9, содержащая следующие поля:

- client_dk – уникальный номер клиента (соответствует client_dk из файла transactions_train.csv);

- колонки с числовыми названиями 27, 32, 41, 45, 67, 73, 81, 88 – это 8 целевых категорий продуктов (описание категорий представлено ниже);значения в колонках с категориями бинарные: 0 – на следующей неделе не было совершено покупок в данной категории, 1 – покупка была совершена.

Описание целевых категорий:

- 27 – спортивные и фитнес-клубы;
- 32 – театры и другие события;
- 41 – ночные клубы и бары;
- 45 – кинотеатры;
- 67 – доставка еды на дом;
- 73 – выставки, художественные музеи, достопримечательности;
- 81 – билеты на спортивные матчи, спортивные клубы и фитнес-центры;
- 88 – луна-парки, океанариумы и цирки.

Ссылка на наборы данных: https://ai-academy.ru/upload/iblock/2ef/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5.zip

Также в соревновательной платформе находился проверочный файл test_target.csv с правильными ответами на тестовый набор данных идентичный по структуре обучающему набору данных с правильными ответами (train_target.csv). Этот файл не доступен участникам соревнования, по нему осуществляется проверка точности решений.

Метрика и формат решений

Качество рекомендательной системы участников считалось на базе метрики $avarageROC_{AUC}$:

$$avarageROC_{AUC} = \sum_{i=1}^{8} \frac{ROC_{AUCi}}{8}$$

где ROC_{AUCi} – усредненный ROC_{AUC} по каждой из 8-ми категорий (т.е. по каждой категории вначале считается ROC_{AUC} по всем клиентам, а затем они суммируются и усредняются на количество категорий).

Участникам нужно было подготовить файл test.csv, имеющий структуру, идентичную train_target.csv (25 000 Клиентов с client_dk и 8 категорий – итого 9 столбцов), заполнив для каждого клиента вероятности покупки в каждой из указанных выше категорий.

Пример выходного файла:

```
client_dk,cat_27,cat_32,cat_41,cat_45,cat_67,cat_73,cat_81,cat_88
12671,0.1,0.2,0.3,0.1,0.1,0.0,0.05,0.15
14015,0.0,0.4,0.3,0.1,0.1,0.1,0.0,0.0
1500,0.0,0.1,0.0,0.5,0.0,0.4,0.0,0.0
4934,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1
11405,0.5,0.1,0.0,0.0,0.0,0.1,0.2,0.1
...
```

Текущий рейтинг формировался автоматически при отправке файла решения на платформу соревнования на основании подсчета метрики $avarageROC_{AUC}$ по публичной части данных (50% файла test_target.csv), позиция в рейтинге отображала лучшее из всех отправленных командой решений.

Для итоговой оценки от каждой команды из двух финальных решений было выбрано лучшее по качеству по приватной части данных (50% данных test_target.csv). Решения, которые участвовали в итоговом оценивании (финальные) команда выбирала самостоятельно в личном кабинете соревновательной платформы.

Рейтинг содержал следующие поля:

- место команды в рейтинге;
- название команды;
- состав команды;
- финальная точность;
- дата и время загрузки последнего решения;
- количество попыток.

Рейтинг финальной задачи

#	Команда	Финальная точность	Последнее решение	Попыток
1	no shakeup allowed	0,8388	06 марта 2020, 16:55	51
2	RoadToBananaDevs	0,8310	06 марта 2020, 16:57	46
3	pudges	0,8309	06 марта 2020, 16:56	52
4	Команда А	0,8306	06 марта 2020, 16:59	65
5	Летающий Пирожок.	0,8304	06 марта 2020, 16:55	72
6	(не)помойная братва	0,8301	06 марта 2020, 17:01	118
7	Адекватное название	0,8299	06 марта 2020, 17:02	64

Человеческий и личный фактор в задачах целиком исключен, вся оценка проводилась в автоматическом режиме.

Пример кода модели, решающей поставленную задачу (https://drive.google.com/file/d/16xefoCu4qQ4arBSUtTNQXG2XCdxdpArl/view?usp=sharing):

Импортируем нужные библиотеки.

```python
import pandas as pd
from sklearn.metrics import roc_auc_score
from sklearn.tree import DecisionTreeClassifier
import numpy as np
```

Распакуйте архив с данными в папку,где находится этот jupyter notebook (baseline.ipynb). У вас будет папка data, содержащая необходимые файлы.

Ваша задача предсказать вероятность совершения покупки Клиентом в определенных 8 категориях в следующие 7 дней, чтобы под них Банк смог направить релевантный контент (подборки). Для подготовки предсказания у вас есть данные по транзакциям 50 000 Клиентов в течение года, из которых 25 000 – в обучающей выборке и 25 000 – в тестовой. Ваша задача для тестовой выборки рассчитать для каждой из целевой категории вероятность совершения покупки в ней на следующей неделе.

Данные по транзакциям находятся в файле **transactions_train.csv**. Информация о покупках по категориям для исследуемой недели находится в файле **train_target.csv**.

Считаем данные по транзакциям и правильные ответы.

```
transactions_train=pd.read_csv('transactions_train.csv')
train_target=pd.read_csv('train_target.csv')
```

Посмотрим на данные.

```
transactions_train.head(5)
```

- client_dk - уникальный идентификатор клиента
- trans_date - дата совершения транзакции
- amount - сумма транзакции
- small_group - категория покупки

```
train_target.head(5)
```

- client_dk - уникальный идентификатор клиента, соответствует полю client_dk из транзакций
- числовые названия колонок - это 8 категорий продуктов. Их названия (числа) соответствуют значениям в колонке small_group из данных по транзакциям. Значения в этих колонках бинарные,т.е. 0 - в исследуемую неделю не было совершено покупки в данной категории, 1 - покупка была совершена. Например, клиент с номером 34110 (первая строчка) купил товар из категории 45, а по остальным категориям покупок не совершал.

Далее представлен простой вариант решения задачи. Вы можете решать соревнование используя совершенно другой подход.

Посчитаем по каждому клиенту самые простые аггрегационные признаки.

```
agg_features = transactions_train.groupby('client_dk')['amount'].agg(['mean', \
    'max', 'min', 'std', 'sum', 'count']).reset_index()
agg_features.head(5)
```

Посчитаем для каждого клиента количество транзакций по каждой категории.

```
counter_df_train = transactions_train.groupby(['client_dk', \
    'small_group'])['amount'].count()
cat_counts_train = counter_df_train.reset_index().pivot(index='client_dk', \
    columns='small_group', values='amount')
cat_counts_train=cat_counts_train.fillna(0)
cat_counts_train.columns = ['small_group_' + str(i) for i in
↪  cat_counts_train.columns]
cat_counts_train.head()
```

Далее соединим все файлы в один датафрейм с таргетом.

```
train=pd.merge(train_target,agg_features,on='client_dk')
train=pd.merge(train,cat_counts_train.reset_index(),on='client_dk')
train.head()
```

Теперь подгрузим тестовые данные для того, чтобы сделать предсказание. Проделаем с ними те же самые манипуляции, как и с обучающими данными.

```
transactions_test=pd.read_csv('transactions_test.csv')
```

Также загрузим id тестовых клиентов, по которым нужно сделать предсказание.

```
test_id=pd.read_csv('test.csv')
agg_features_test=transactions_test.groupby('client_dk')['amount'].agg(['mean',
↪  'max', 'min', 'std', 'sum', 'count']).reset_index()
counter_df_test=transactions_test.groupby(['client_dk',
↪  'small_group'])['amount'].count()
cat_counts_test=counter_df_test.reset_index().pivot(index='client_dk',
↪  columns='small_group',values='amount')
cat_counts_test=cat_counts_test.fillna(0)
cat_counts_test.columns=['small_group_'+str(i) for i in cat_counts_test.columns]
cat_counts_test.head()
test=pd.merge(test_id[['client_dk']],agg_features_test,on='client_dk')
test=pd.merge(test,cat_counts_test.reset_index(),on='client_dk')
common_features=list(set(train.columns).intersection(set(test.columns)))

X_train=train[common_features]
X_test=test[common_features]
```

В этом бэйзлайне мы будем использовать простой подход - предсказывать покупки в каждой категории независимо. То есть в цикле модель обучается на отдельную категорию как на зависимую переменную, и пытается предсказать наличие покупки в этой определенной категории для теста. В итоге у нас получается 8 задач бинарной классификации.

Важно: Такой подход не претендует на звание лучшего, вы вольны придумать свой алгоритм решения, который, вполне вероятно, окажется лучше.

```
#В словарь будем записывать предсказания модели
results_tree = {}
#Цикл со второго элемента, потому что первой колонкой идет идентификатор клиента
↪
for q in train_target.columns[1:]:
    print('train product '+str(q))
    curr_target_train = train_target.loc[:,q]
    model = DecisionTreeClassifier(random_state=42)
    model.fit(X_train.fillna(0).values,curr_target_train.values)
    #Сделаем предсказание
    pred = model.predict_proba(X_test.fillna(0).values)[:,1]
    results_tree[q] = pred
```

Такое решение дает на публичном лидерборде качество 0.6023.

Подготовим файл для отправки в систему

```
submission = pd.DataFrame(data = np.zeros((25000, 8)), columns =
↪  train_target.columns[1:], index = test_id['client_dk'].values)
for q in results_tree:
    submission[q] = results_tree[q]

submission.index.name = 'client_dk'
submission.columns = ['cat_27', 'cat_32', 'cat_41', 'cat_45', 'cat_67',
↪  'cat_73', 'cat_81', 'cat_88']
```

Сохраняем прогноз на диск в папку submissions. Имя прогноза соответсвует дате и времени его создания, закодированными с помощью timestamp.

```
import time
import os

current_timestamp = int(time.time())
submission_path = 'submissions/{}.csv'.format(current_timestamp)

if not os.path.exists('submissions'):
    os.makedirs('submissions')

print(submission_path)
submission.to_csv(submission_path, index=True)
```

Теперь все готово! Можно отправлять решение.

Критерии

Критерии определения победителей и призеров заключительного этапа

Первый отборочный этап

В первом отборочном этапе участники решали задачи по двум предметам: математика и информатика, в каждом предмете максимально можно было набрать 100 баллов. Для того, чтобы пройти во второй этап участники должны были набрать в сумме по обоим предметам не менее 75 баллов, независимо от уровня.

Второй отборочный этап

В ходе 2 этапа участники загружали решения на платформу, где осуществлялась автоматическая проверка присланных решений, в результате которой формировался рейтинг (https://onti.ai-academy.ru/leaderboard). Первые 100 участников рейтинга, чья модель машинного обучения наиболее точно справилась с поставленной задачей, были определены финалистами. В случае, если кто-то из участников отказывался, приглашался следующий в рейтинге. Общее число финалистов – 100 участников.

Заключительный этап

Индивидуальный предметный тур:

Математика – максимально возможный балл за все задачи - 100 баллов;

Информатика – максимально возможный балл за все задачи - 100 баллов.

Командный практический тур:

Командный практический тур – это решение задачи по машинному обучению командой из двух участников.

Команды были ограничены 100 попытками в день. Попытка – возможность загрузить ответ и узнать точность решения.

Максимально количество баллов, которое можно было набрать за решение командной задачи - 100.

Данный балл присваивался команде, занявшей первое место в рейтинге. Для определения баллов остальных команд основным параметром формулы являлась финальная точность - результат автоматической проверки присланных командами решений, который определялся путем сравнения результатов предсказания модели команд с реальными данными. Так как различия в финальной точности незначительные (сотые доли процента), в качестве дополнительного параметра было введено место в рейтинге: чем выше в рейтинге находилась команда, тем тем больший коэффициент она получала при расчете баллов.

Таким образом, с учетом того, что в соревнованиях участвовало 50 команд, результирующий балл команды равен:

$$\frac{\textit{финальная точность} \times (51 - \textit{место команды в рейтинге})}{\textit{максимальная финальная точность среди всех команд} \times 50} \times 100$$

Личный итоговый балл каждого участника вычислялся по формуле:

$$\textit{Итоговый балл} = \textit{балл предметного тура по математике} \times 0,1+$$

$$+\textit{балл предметного тура по информатике} \times 0,1 + \textit{балл за решение командной задачи} \times 0,8.$$

Критерий определения победителей и призеров (независимо от класса):

Построен общий рейтинг, где 8-9-ые классы участвовали на общих основания с 10-11 классами. С начала рейтинга были выбраны 8 победителей и 17 призеров (первые 25% участников рейтинга становятся победителями и призерами – первые 8% участников рейтинга становятся победителями).

Критерий определения победителей и призеров (независимо от уровня):

Категория	**Количество баллов**
Победители	84,56 и выше
Призеры	От 68,44 до 81,69