Задача A: Перестройка

В команде Тинькофф произошла реорганизация — объединились два отдела. Новые коллеги хотят сидеть рядом, поэтому офису требуется ремонт.
Прежние места каждого отдела имели форму прямоугольника. Новая площадка должна быть квадратной, а также содержать предыдущие места, т.к. некоторые разработчики очень привязаны к ним. К сожалению, размеры офиса ограничены, поэтому зона должна иметь минимальную площадь. Строители пытаются посчитать, сколько материала им понадобится для ремонта. Для этого им нужно знать итоговую площадь обновленной площадки. Пожалуйста, помогите им.

Решение

Чтобы покрыть оба прямоугольника, нужно покрыть все точки с x координатой из отрезка $[x_{min};\ x_{max}]$, где x_{min} — минимальная x координата среди всех вершин прямоугольников. То же верно и для y координаты. Значит, новая зона должна покрывать прямоугольник $[x_{min};\ x_{max}] \times [y_{min};\ y_{max}]$. Для того, чтобы покрыть такой прямоугольник квадратом, достаточно взять сторону квадрата, равной наибольшей стороне этого прямоугольника.

Задача B: Победители

Аня — координатор стажировок в Тинькофф. Она хочет нанять самых сильных олимпиадников.
Для того чтобы понять, кто же лучший, Аня решила проанализировать результаты командной олимпиады за последние N лет. Она знает все команды, занявшие первое место. Каждая команда задается тройкой имен, причем их порядок не важен, то есть записи $ANTON\ BORIS\ CHRIS$ и $BORIS\ ANTON\ CHRIS$ задают одну и ту же команду.
Ане нужны лучшие из лучших, поэтому она хочет знать, какое максимальное число раз побеждала команда в одном и том же составе. Вы дружите с Аней, поэтому согласились ей помочь.

Решение

Для начала отсортируем каждую команду лексикографически, так как порядок членов команды не имеет значения для состава. Теперь для каждой команды можно пробежаться по всем остальным и сравнить составы, а при совпадении увеличивать счетчик вхождения текущей команды. После подсчета промежуточного ответа нужно проверить, больше ли он глобального ответа на задачу. Это решение имеет квадратичную асимптотику, что укладывалось в ограничения.

Задача C: Подписки

Один из ваших знакомых стал стажером-аналитиком в команде Тинькофф. У него есть данные за n дней о том, как клиенты подключают и отменяют подписки на один из сервисов. По странному стечению обстоятельств дни чередуются: в первый день подписки только подключались, во второй только отменялись и т.д., т.е. каждый день число приобретенных подписок составляет $(-1)^{i-1}a_i$, где a_i — число подключенных или отключенных подписок за i-й день.
Во время анализа данных ваш знакомый задался вопросом, увеличилась бы прибыль компании, если бы данные за какие-то дни поменялись местами. В качестве первого этапа он решил поменять местами не более двух дней.
Проверьте, сможете ли вы стать стажером-аналитиком в Тинькофф. Для этого предлагаем вам посчитать максимально возможное значение прибыли, которую можно получить, поменяв местами данные за не более, чем два дня. Стоимость одной подписки можно считать равной 1.

Решение

Рассмотрим вклад чисел, стоящих на четных и нечетных местах. Заметим, что для увеличения максимальной прибыли нужно заменить максимальное число с четным индексом (отмененные подписки) заменить на минимальное число с нечетным индекстом (подключенные подписки). Причем менять нужно только в том случае, если оно не ухудшает итоговый результат (именно поэтому можно вообще не делать обменов).
Докажем, почему всегда выгодно так делать. Для начала заметим, что менять числа на местах с одинаковой четностью нет смысла, так как общий результат суммирования прибыли никак не изменится. Далее предположим, что в оптимальном решении меняются числа с индексами i и j, а в нашем — с индексами k и l, причем i и k нечетные, а j и l четные. Тогда если вместо a_i менять a_k, то мы улучшим оптимальный овтет на $+a_i - a_k$, что ≥ 0, так как элемент на позиции k — минимальный среди стоящих на нечетном месте. То же верно и для другой пары индексов: если менять не j-й элемент, а l-й, то ответ увеличится на $+a_l - a_j$, что ≥ 0, так как a_l — максимальный среди стоящих на четном месте. Следовательно, наше решение не хуже, чем оптимальное, то есть является таковым.

Задача D: Первое задание

Вы прошли на стажировку и получили свое первое задание.
Ваш куратор попросил прочитать и обработать файл конфигурации, состоящий из строк трех типов:

- { — начало блока;
- } — конец блока;

- $variable =< value >$ — присваивание в переменную с именем $variable$ значения $value$.

$value$ может быть двух типов:

- число, по модулю не превосходящее 10^9;
- название переменной, чье значение нужно присвоить.

В файле нет пробелов, и каждая операция записана в отдельной строке. Имена всех переменных состоят из не более чем 10 строчных латинских букв $(a, \ldots, z)$. Изначально все переменные имеют значение 0. Как только встречается выражение присваивания, то необходимо сразу же его выполнить. Это значение сохраняется до конца блока (если, конечно, не будет перезаписано), и после конца блока возвращается старое значение переменной.
Куратор хочет проверить, насколько хорошо вы анализируете код, поэтому попросил вас вывести значение переменной $variable_1$ после каждого присваивания вида $variable_1 = variable_2$.

Решение

Будем читать файл построчно и поддерживать для каждой переменной стек значения, которые были присвоены. Кроме того, для каждого уровня вложенности блоков будем поддерживать множество переменных, которые были изменены на этом уровне.
При каждом новом присваивании нужно посмотреть, менялось ли уже значение этой переменной в этом блоке. Если нет, то нужно добавить на стек новое значение, а если да, то перезаписать его. При выходе из блока для всех переменных, которые были на нем изменены, нужно убрать последнее присвоенное значение, то есть последнее число из стека для этой переменной.
При таких операциях мы всегда сможем узнать текущее значение переменной, посмотрев на вершину стека.

Задача E: Префиксы

Пройдя тестовое задание от куратора из предыдущей задачи, вы получили новое. На этой раз вам нужно улучшить систему поиска карточек в бухгалтерии Тинькофф.
Всего у нас работает n людей. Каждый человек определяется своей фамилией, состоящей из строчных букв латинского алфавита $(a, \ldots, z)$. К сожалению, бумажные записи со временем становятся нечитаемыми, так как конец фамилии стирается, но команда бухгалтерии отлично знает систему хранения карточек и умеет находить любого сотрудника по префиксу его фамилии.

Для более быстрой работы дополнительно требуется знать k-го в лексикографическом порядке человека среди всех с заданным префиксом. Задачу быстрого поиска такого человека и поставил перед вами куратор.

Решение

Отсортируем все фамилии. После этого для ответа на запрос нужно было бы искать k-е слово с подходящим префиксом. Например, это можно сделать при помощи линейного прохода, но тогда бы решение работало за $\mathcal{O}(nq\sum|s_i|)$, что не укладывается в ограничения по времени, так как всего нужно сделать q запросов.
Так как все фамилии с подходящим префиксом после сортировки лежат подряд, то для ускорения решения можно воспользоваться бинпоиском для нахождения первого слова с заданным префиксом. Далее необходимо посмотреть на $(found + k_i)$-е слово в отсортированном порядке и проверить, что оно подходит, то есть имеет заданный префикс. Это решение тратит $\mathcal{O}(\log n|s_i|)$ на запрос, что удовлетворяет ограничениям.

Задача F: Поехали

В офисе Тинькофф есть несколько лифтов для минимизации времени ожидания и ускорения перемещения по зданию. У лифтов есть особенность: i-й лифт едет только с этажа s_i до этажа f_i без промежуточных остановок. По задумке строителей лифты везут пассажиров только вверх (вниз все ходят по лестницам).
В первый день стажировки вы решили воспользоваться этими особенностями, а именно прокатиться на максимальном числе лифтов подряд, составив цепь. Цепью вы называете последовательность лифтов, для которых для любых двух лифтов, имеющих в цепи номера i и $i+1$ выполняется условие $f_i = s_{i+1}$, то есть между двумя лифтами вам не нужно пользоваться лестницей, чтобы добраться от одного до другого.
Определите максимально возможную длину цепи лифтов, на которых вам удастся прокатиться.

Решение

Будем решать эту задачу с помощью динамического программирования.
Отсортируем лифты по возрастанию их конечного этажа. Посчитаем динамику dp_k — максимальную длину цепочки, позволяющей доехать до k-о этажа. Рассмотрим i-й лифт. Если мы возьмем его в ответ, то для получения цепочки предыдущий лифт должен иметь конечный этаж, равный стартовому для i-о лифта. Значит, надо пробовать улучшать ответ для высоты r_i ответом для высоты l_i: $dp_{r_i} = max(dp_{r_i}, dp_{l_i} + 1)$. Динамика будет считаться правильно из-за сортировки лифтов по возрастанию конечного этажа.

Заметим, что из-за ограничений на высоту лифтов (они могут быть до 10^9), перед подсчетом значения стартового и конечного этажей необходимо сжать или воспользоваться *map*-ом.

Задача G: Полки

После работы ваш коллега, стажер Павел, решил зайти в магазин. Как и вы, Павел — разработчик, поэтому каждое свое действие он выполняет по алгоритму в строгой последовательности.
Супермаркет для Павла — прямая с полками. На каждой полке стоят товары одной категории, а каждая полка помечена какой-то строчной буквой латинского алфавита $(a, \ldots, z)$, то есть весь супермаркет можно представлять как строку s.
Павел хочет взять по одному товару с каждой полки в каком-то порядке. Для этого он делает две операции:

- Взять товар с текущей полки и положить в корзину, если он этого не сделал ранее.
- Передвинуться к следующей полке. Если он стоял у последней полки, он возвращается к первой.

Павел любит порядок и хочет складывать товары в отсортированном порядке, а именно сначала он хочет взять по одному товару с полок с буквой a, если они есть, затем — с буквой b и так далее до z. У Павла был тяжелый день, он хочет домой, поэтому планирует закончить с покупками как можно быстрее. Для этого он решил брать товары не со всех полок, а только с какого-то подотрезка, т.е. рассматривать все полки с l-й по r-ю.
Пожалуйста, помогите Павлу быстрее попасть домой и посчитайте, сколько передвижений, то есть операций второго типа, ему нужно будет сделать.

Решение

Для начала рассмотрим медленное решение задачи. Для каждого запроса будем поддерживать двумерный вектор, где для каждой буквы будут записаны все ее вхождения. Дальше остается только смоделировать движение Павла: для первой буквы нужно в отсортированном порядке пройти все буквы, а для всех следующий нужно сначала найти ближайшую справа букву. Полученное решение работает за $\mathcal{O}(|s|)$ на запрос. При числе запросов и длины супермаркета до 10^5 такое решение не укладывается во временные ограничения.
Для ускорения заметим, что для каждой буквы нас интересуют не все ее вхождения, а только последнее на подотрезке и последнее слева от текущей позиции Павла. Пояснение — если у нас нет буквы слева от Павла, то достаточно дойти до последнего вхождения на подотрезке, а иначе нужно идти до его конца, а потом возвращаться до этого вхождения. Необходимые позиции можно поддерживать при помощи массива предыдущих букв

$prev[position][letter]$, причем его достаточно предпосчитать в начале, а далее только использовать. В итоге получаем $\mathcal{O}(26|s|)$ на предпосчет и $\mathcal{O}(1)$ на запрос.

Задача H: Покупатели

В Тинькофф решили освоить новую нишу, а именно открыть внутренний стартап по продаже доменов. Каждый домен представляет из себя строку, состоящую только из букв S, T, A, R.
Каждому покупателю хочется не просто продать строку, а продать нужную строку, а именно, пусть у каждого покупателя есть строки P и Q. Тогда ему подходят только строки, у которых первые $|P|$ символов совпадают с P и последние $|Q|$ символов совпадают с Q.
Для начала было закуплено n доменов для m покупателей. Теперь хочется понять, хватит ли их для всех покупателей. Чтобы получить ответ на этот вопрос, посчитайте число подходящих для каждого покупателя строк.

Решение

Для начала рассмотрим более простую версию задачи: предположим, что в запросе был указан только префикс строки. То есть запрос заключается в том, что нам нужно вывести количество строк из словаря, у которых первые $|P_i|$ символов совпадают с P_i. Для решения такой задачи воспользуемся бором. Запишем все S_i (домены) и все P_j в бор. Теперь заметим, что утверждение «строка S является префиксом строки T» эквивалентно утверждению «вершина бора x, соответствующая строке S, является предком вершины v, соответствующей строке T». Это легко показать из устройства бора. То есть наша задача свелась к подсчету количества вершин в поддереве.
Для решения исходной задачи будем пользоваться эйлеровым обходом полученного дерева, а именно посчитаем времена входа и выхода для каждой вершины, соответствующей какой-либо строке. Тогда каждый запрос покупателя представим в качестве отрезка, а каждую строку из набора — в виде точки (можно, например, сопоставлять время входа). Упрощенная задача же заключается в поиске количества точек внутри каждого отрезка.
Те же рассуждения можно проделать для развернутых строк S_i и Q_j. Тогда исходная задача сводится к поиску количества точек в каждом прямоугольнике. Это можно сделать, например, при помощи многомерного сжатого дерева Фенвика.