

Задача А. Место встречи изменить нельзя

Имя входного файла: `rendezvous.in`
Имя выходного файла: `rendezvous.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даны N точек. Найдите такие две из них, что расстояние между ними минимально.

Формат входных данных

Первая строка входного файла содержит целое число N ($2 \leq N \leq 100\,000$) — количество точек. Каждая из следующих N строк содержит пару целых чисел X и Y , разделённых пробелом, — координаты ($-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$). Все точки различны.

Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

Пример

| <code>rendezvous.in</code> | <code>rendezvous.out</code> |
|----------------------------|-----------------------------|
| 4 | 0 0 |
| 0 0 | 0 1 |
| 0 1 | |
| 1 1 | |
| 1 0 | |

Задача В. Соединение и разъединение

Имя входного файла: `connect.in`
Имя выходного файла: `connect.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Примеры

| <code>connect.in</code> | <code>connect.out</code> |
|-------------------------|--------------------------|
| 5 11 | 5 |
| ? | 1 |
| + 1 2 | 1 |
| + 2 3 | 2 |
| + 3 4 | |
| + 4 5 | |
| + 5 1 | |
| ? | |
| - 2 3 | |
| ? | |
| - 4 5 | |
| ? | |

Задача С. Большие Диаграммы Юнга

Имя входного файла: multiply.in
Имя выходного файла: multiply.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Яша и Саша — два роскошных брата, которые любят дарить друг другу подарки на день рождения. И вот, на очередной день рождения, Яша подарил Саше на день рождения Диаграмму Юнга. Саша решил посчитать число клеток в этой диаграмме. Но Саша опаздывал на матч по футболу, поэтому посчитал только число клеток в первой строке и первом столбце. И теперь ему интересно, какое максимальное по модулю число клеток может быть в этой диаграмме.

Формат входных данных

В первой строке входных данных находится одно целое число — размер первой строки диаграммы. Во второй строке находится другое целое число — размер первого столбца диаграммы. Длины этих чисел не превосходит двухсот пятидесяти тысяч символов. Размеры строк этой диаграммы могут быть отрицательными.

Формат выходных данных

В единственной строке выведите единственное число — максимальный по модулю размер диаграммы.

Примеры

| multiply.in | multiply.out |
|-------------|--------------|
| 2 | 4 |
| 2 | |

Задача D. Ciel и гондолы

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Лиса Ciel зашла в парк аттракционов. И вот, она в очереди на колесо обозрения. В очереди стоит n людей (хотя нет, скорее лис): мы будем считать, что первая лиса стоит в начале очереди, а n -я лиса стоит в хвосте очереди.

Всего имеется k гондол, мы распределяем лис по гондолам следующим образом:

- Когда подплывает первая гондола, q_1 лис переходят из начала очереди в подплывшую гондолу.
- Затем, когда подплывает вторая гондола, q_2 лис из начала оставшейся очереди переходит в эту гондолу.
- ...
- Оставшиеся q_k лис идут с последней (k -ю) гондолу.

Обратите внимание, что числа q_1, q_2, \dots, q_k должны быть положительными. Из условия следует, что $\sum_{i=1}^k q_i = n$ и $q_i > 0$.

Вы знаете как лисам не хочется задерживаться в гондолах с незнакомцами. Итак, Ваша задача — найти оптимальный способ размещения (то есть определить оптимальную последовательность q), чтобы угодить всем. Для каждой пары лис i и j задано значение u_{ij} , обозначающее степень незнакомости. Можете считать, что $u_{ij} = u_{ji}$ для всех i, j ($1 \leq i, j \leq n$) и что $u_{ii} = 0$ для всех i ($1 \leq i \leq n$). Тогда значение незнакомости в гондоле определяется как сумма значений незнакомости между всеми парами лис, которые находятся в этой гондоле. Общее значение незнакомости определяется как сумма значений незнакомости по всем гондолам.

Помогите лисе Ciel найти минимальное возможное значение общей незнакомости при некотором оптимальном распределении лис по гондолам.

Формат входных данных

В первой строке даны два целых числа n и k ($1 \leq n \leq 4000$ and $1 \leq k \leq \min(n, 800)$) — количество лис в очереди и количество гондол. В следующих n строках записано по n целых чисел — матрица u , ($0 \leq u_{ij} \leq 9$, $u_{ij} = u_{ji}$ и $u_{ii} = 0$).

Пожалуйста, используйте методы быстрого чтения (например, для Java используйте `BufferedReader` вместо `Scanner`).

Формат выходных данных

Выведите целое число — минимальное возможное значение общей незнакомости.

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 5 2 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 | 0 |
| 8 3 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 | 7 |

Замечание

В первом примере можно распределить лис вот так: 1, 2 идут в одну гондолу, 3, 4, 5 идут в другую гондолу.

Во втором примере оптимальное распределение таково: 1, 2, 3 | 4, 5, 6 | 7, 8.

Задача Е. Легенды и мифы Южного Бутово

| | |
|-------------------------|--------------|
| Имя входного файла: | police.in |
| Имя выходного файла: | police.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

В 3141 году Бутово превратилось в очень опасный район, полный убийц и прочих жуликов. Настолько опасный, что там стало страшно перемещаться даже на танке!

Напомним, что Бутово состоит из N проспектов, идущих по направлению с севера на юг, и M улиц, идущих с востока на запад. Каждый проспект пересекается с каждой улицей ровно на одном перекрестке. Перемещение вдоль проспекта или улице на высокой скорости безопасно, а поворот где-либо, наоборот, очень опасен, так как для этого приходится снизить скорость, и в тот же миг вас начинают атаковать группировки местных жителей.

Управление полицией Бутово решило немного исправить ситуацию. Руководители управления решили, что можно поставить несколько постов на некоторых перекрестках, чтобы на них жители могли спокойно поворачивать, не опасаясь быть атакованными, так как полиция уже взяла на вооружение новейшие винтовки «Мушкет-1812» и, в случае чего, сможет защитить честь и достоинство каждого законопослушного гражданина.

К сожалению, именно в этом году начальник управления решил построить себе новый элитный особняк, поэтому управление решило потратить как можно меньше денег на строительство новых постов. Однако, чтобы все выглядело чистенько, необходимо построить посты так, чтобы можно было добраться от любой точки любой улицы до любой точки любой другой, поворачивая лишь на специально оборудованных постах. Иначе нагрянет проверка и всех уволят.

На этом проблемы Бутово не закончились. Некоторые районы Бутово являются более опасными, и посты, которые будут построены в более опасных районах, стоят дороже. У каждого района есть центр, который находится возле некоторого перекрестка. Любой другой перекресток принадлежит району, центр которого является ближайшим к перекрестку. Расстояние между перекрестками измеряется с помощью так называемого Бутовского расстояния: расстояние между двумя различными перекрестками — это минимальное количество промежуточных перекрестков, которое нужно преодолеть, чтобы попасть из одного в другой, плюс один. Если же существует несколько районных центров, находящихся на минимальном расстоянии от некоторого перекрестка, то на этом перекрестке постоянно происходят стычки двух авторитетов, поэтому на этом перекрестке построить пост невозможно.

Вам требуется, зная расположение районов Бутово, определить минимальную стоимость постройки постов так, чтобы можно было безопасно добраться от любой точки любой улицы до любой точки любой другой.

Формат входных данных

Первая строка входного файла содержит три числа N , M , R — количество проспектов, количество улиц и количество районов в Бутово, соответственно ($2 \leq N, M \leq 500$, $1 \leq R \leq 1000$). Следующие R строк содержат по три целых числа — номер проспекта и номер улицы, на которых расположен центр соответствующего района, и стоимость постройки поста в этом районе. Проспекты и улицы нумеруются с единицы, стоимость постройки всюду положительна и не превосходит одной тысячи. Никакие два района не имеют совпадающих центров.

Формат выходных данных

На первой строке выходного файла выведите два числа: количество перекрестков K , на которых стоит разместить посты, и найденную минимальную суммарную стоимость постройки. На следующих K строках должны содержаться описания найденных перекрестков: номер проспекта и номер улицы, на пересечении которых расположен соответствующий перекресток. Если посты нужным образом разместить невозможно, выведите в выходной файл единственное число -1

Примеры

| police.in | police.out |
|-----------------------------|--|
| 4 3 2 3 1 200 1 3 150 | 6 1050 2 3 1 3 1 2 4 1 4 2 3 2 |
| 3 3 2 1 2 200 3 2 150 | -1 |

Задача F. Наибольший тандемный повтор

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам дана строка s . Найдите длину наибольшего тандемного повтора, являющегося подстрокой s .

Формат входных данных

В первой строке дано одно целое положительное число t — количество тестов.

В каждой из следующих t строк дается одна строка s , состоящая из символов с кодами от 33 до 126 включительно ($|s| \leq 10^5$).

Суммарная длина строк не превосходит $3 \cdot 10^6$.

Формат выходных данных

Для каждого теста выведите в новой строке одно число — максимальную длину тандемного повтора, являющегося подстрокой строки s .

Примеры

| стандартный ввод | стандартный вывод |
|--------------------|-------------------|
| 4 | 0 |
| abcd | 10 |
| ATTCGATTCGATTCG | 2 |
| ggwp | 4 |
| 456765654348298742 | |