

01A. Удаление клеток

2 секунды, 64 мегабайта

Из прямоугольного листа клетчатой бумаги (M строк, N столбцов) удалили некоторые клетки. На сколько кусков распадется оставшаяся часть листа? Две клетки не распадаются, если они имеют общую сторону.

Входные данные

В первой строке находятся числа M и N ($1 \leq M, N \leq 100$), в следующих M строках - по N символов. Если клетка не была вырезана, этому соответствует символ О, если вырезана - точка.

Выходные данные

Вывести одно число - на сколько кусков распадется лист.

входные данные
5 10 00..00000. .0.0.0.... 000..00.0. ..00.....0 .000.00000
выходные данные
5

01B. Кратчайший путь

2 секунды, 256 мегабайт

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Входные данные

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя целыми числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$.

Выходные данные

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите «Unreachable».

входные данные
2 1 1 2 1 2 -10
выходные данные
-10

входные данные
2 1 2 1 1 2 -10
выходные данные
Unreachable

01C. Современное искусство

2 секунды, 256 мегабайт

Серёжа — современный художник, поэтому в своих композициях он использует 3D-принтер. В данный момент Серёжа придумал идею следующей композиции. Он напечатает n разноцветных пластиковых брусков и попытается расположить эти бруски горизонтально один над одним (бруски можно считать горизонтальными отрезками на плоскости, имеющими ненулевую длину и лежащими на попарно различных прямых, параллельных оси абсцисс), но сделать это следует с учетом соотношений цветов (Серёжа же всё-таки художник!). Соотношения бывают двух видов: соотношение предшествования (тип А) и соотношение пересечения (тип В).

Если для двух брусков i и j есть соотношение типа А, то брусок i должен лежать строго левее бруска j (то есть координата по оси абсцисс любой точки бруска i должна быть строго левее координаты по оси абсцисс любой точки бруска j).

Соотношение типа В между брусками i и j означает, что данные бруски должны иметь хотя бы по одной точке, не являющейся границей бруска, проекции которых на ось абсцисс совпадают.

Если между двумя брусками нет ни соотношения А, ни соотношения В, то они могут располагаться друг относительно друга в любом порядке.

Помогите Серёже найти идеальные координаты по оси абсцисс для своих брусков или определить, что это невозможно. В случае успеха своей композиции он, может быть, даже поделится с вами гонораром!

Входные данные

В первой строке задано три целых числа через пробел: n , a и b ($1 \leq n \leq 10^5$, $0 \leq a, b \leq 10^5$) — количество брусков, а также количество ограничений первого и второго типа соответственно.

В следующих a строках указаны пары целых чисел $f_a[i]$ и $s_a[i]$ ($1 \leq f_a[i], s_a[i] \leq n, f_a[i] \neq s_a[i]$) — номера брусков, для которых должно выполняться соотношение типа А.

В следующих b строках указаны пары целых чисел $f_b[i]$ и $s_b[i]$ ($1 \leq f_b[i], s_b[i] \leq n, f_b[i] \neq s_b[i]$) — номера брусков, для которых должно выполняться соотношение типа В.

Выходные данные

Если расположения брусков, удовлетворяющего данным требованиям, не существует, выведите в единственной строке «NO» (без кавычек).

В противном случае в первой строке выведите «YES» (без кавычек). Далее, для каждого из n брусков на отдельной строке выведите два целых числа l_i и r_i ($0 \leq l_i < r_i \leq 10^9$) — координаты левого и правого концов i -го бруска по оси абсцисс в удовлетворяющем всем требованиям расположении.

входные данные
4 2 1 1 2 2 3 3 4
выходные данные
YES 0 1 2 3 4 6 5 7

входные данные
3 2 0 1 2 2 1
выходные данные
NO

02A. Противопожарная безопасность

2 секунды, 256 мегабайт

В Марксе n домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Марксе участились случаи пожаров. В связи с этим жители решили построить в городе несколько пожарных станций. Но возникла проблема: едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако, возвращаясь с задания машина обязана следовать правилам дорожного движения (жители Маркса свято чтут эти правила!).

Ясно, что где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой выехала. Но строительство станций стоит больших денег, поэтому на совете города было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

Входные данные

В первой строке входного файла задано число n ($1 \leq n \leq 3\,000$).

Во второй строке записано количество дорог m ($1 \leq m \leq 100\,000$).

Далее следует описание дорог в формате $a_i\ b_i$, означающее, что по i -й дороге разрешается движение автотранспорта от дома a_i к дому b_i ($1 \leq a_i, b_i \leq n$).

Выходные данные

В первой строке выведите минимальное количество пожарных станций K , которые необходимо построить. Во второй строке выведите K чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

входные данные
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5
выходные данные
2 4 5

02B. Экскурсия

2 секунды, 512 мегабайт

Группа из n человек решила поехать на экскурсию. В процессе экскурсии можно заехать в некоторые из m городов.

Экскурсовод попросила каждого человека высказать свои пожелания по поводу посещения городов. Каждый человек может про какие-то города заявить, что он хочет их непременно посетить, а про какие-то — что точно не хочет.

Группа всегда путешествует вместе. Если группа заезжает в город, то все люди, заявившие, что точно не хотят его посетить, расстраиваются. Если группа не заезжает в город, то расстраиваются все люди, которые заявили, что хотят его непременно посетить.

Экскурсовод понимает, что удовлетворить все пожелания не всегда можно. Она хочет составить план, чтобы каждый человек расстроился не более одного раза.

Помогите экскурсоводу справиться с этой нелегкой задачей и составьте план посещения городов или выясните, что это невозможно.

Входные данные

В первой строке входных данных содержатся три целых числа n , m , k — количество человек, количество городов и количество пожеланий ($1 \leq n, m, k \leq 100\,000$).

В каждой из последующих k строк содержатся по два целых числа a и b , обозначающих пожелания ($1 \leq a \leq n, 1 \leq |b| \leq m$). Если $b > 0$, то человек под номером a хочет посетить город под номером b . Если же $b < 0$, то человек под номером a не хочет посетить город под номером $-b$. Каждое пожелание указано во вводе не более одного раза, ни для кого из участников нет одновременно пожелания посетить и не посещать один и тот же город.

Выходные данные

Если решения не существует, то выведите -1 .

Иначе, в первой строке выходных данных выведите единственное целое число k — количество городов, которые войдут в план.

Во второй строке выведите k целых чисел — номера городов, которые следует посетить. Номера городов можно выводить в любом порядке.

Если существует несколько возможных ответов, можно вывести любой из них. Обратите внимание, что не требуется искать максимальное или минимальное k , можно вывести любой корректный ответ.

входные данные
3 5 6 1 2 1 3 1 -4 2 3 2 4 2 5
выходные данные
3 2 3 5

входные данные
3 3 6 1 -1 1 2 2 -2 2 3 3 -3 3 1
выходные данные
0

02C. Планирование вечеринки

2 секунды, 512 мегабайт

Петя планирует вечеринку, это дело непростое. Одна из главных проблем в том, что некоторые его друзья плохо ладят друг с другом, а некоторые — наоборот. В результате у него есть множество требований, например: «Я приду только если придет Гена» или «Если там будет Марина, то меня там точно не будет».

Петя формализовал все требования в следующем виде:

« $[+-]\text{name1} \Rightarrow [+-]\text{name2}$ », здесь « name1 » и « name2 » — имена двух друзей Пети, « $+$ » означает, что друг придет в гости, « $-$ » — что не придет. Например, выражение «Если Андрея не будет, то Даша не придет» записывается так: « $-\text{andrey} \Rightarrow -\text{dasha}$ ».

Помогите Пете составить хоть какой-нибудь список гостей, удовлетворяющий всем свойствам, или скажите, что это невозможно

Входные данные

В первой строке входного файла записаны числа n и m — число друзей Пети и число условий ($1 \leq n, m \leq 1000$). В следующих n строках записаны имена друзей. Имена друзей состоят из маленьких латинских букв и имеют длину не больше 10. В следующих m строках записаны условия.

Выходные данные

Выведите в первой строке число k — число друзей, которых нужно пригласить. В следующих k строках выведите их имена.

входные данные
3 3 vova masha gosha -vova => -masha -masha => +gosha +gosha => +vova
выходные данные
2 vova masha

входные данные
1 1 vova -vova => +vova
выходные данные
1 vova

входные данные
2 4 vova masha +vova => +masha +masha => -vova -vova => -masha -masha => +vova
выходные данные
-1

03A. Граф

1 секунда, 256 мегабайт

Много лет назад Альфу на день рождения подарили неориентированный граф. Недавно, разбирая вещи, он нашел его, и сильно удивился. Поиграв с ним некоторое время, ему стало интересно: сколько в этом графе есть пар вершин, между которыми нет ребра, а если его провести, в графе появится ровно один новый вершинно-простой цикл?

Входные данные

В первой строке входного файла заданы два целых числа n, m ($1 \leq n, m \leq 10^5$) — число вершин и ребер в графе соответственно.

В следующих m строках заданы пары целых чисел u, v ($1 \leq u, v \leq n$) — ребра графа.

Выходные данные

В единственной строке выходного файла выведите искомое число пар вершин.

входные данные
5 4 1 2 2 3 3 4 4 5
выходные данные
6

входные данные
5 5 1 2 2 3 1 3 3 4 4 5
выходные данные
1

03B. Магнитные подушки

2 секунды, 256 мегабайт

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добираться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

Входные данные

В первой строке входного файла находятся числа N и M — количество небоскребов в городе и количество работающих магнитных подушек соответственно ($3 \leq N \leq 100000$, $1 \leq M \leq 100000$). В каждой из следующих M строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до N . Гарантируется, что имеющиеся воздушные подушки позволяют перемещаться от одного небоскреба до любого другого.

Выходные данные

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

входные данные
3 1 1 2 3
выходные данные
1 1

входные данные
3 2 1 2 3 3 2 1
выходные данные
0

входные данные
5 4 1 2 3 2 4 3 1 2 4 3 5 1
выходные данные
1 4

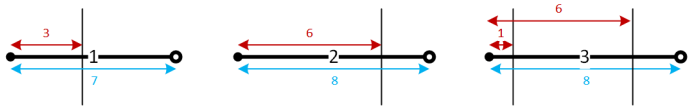
03С. Антенна

1 секунда, 512 мегабайт

Для связи с Землёй членам экспедиции на Марс необходимо собрать антенну. Антенна в разобранном состоянии представляет собой n фрагментов, i -й фрагмент представляет собой штангу длиной s_i сантиметров, на которой закреплены m_i перекладин. Каждый фрагмент содержит хотя бы одну перекладину.

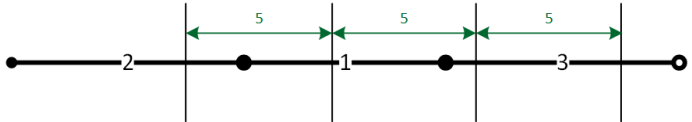
У каждой штанги есть начало, в котором расположен штекер, и конец, в котором расположено гнездо. Любые две штанги можно последовательно соединить, присоединив начало одной к концу другой. Для каждой перекладины известно расстояние от начала её штанги в сантиметрах. Для i -го фрагмента это расстояние может быть от 0 до s_i , значение 0 означает, что перекладина находится непосредственно в начале штанги, значение s_i — что она находится непосредственно в конце штанги. Толщиной перекладин и размерами штекера и гнезда следует пренебречь.

На рисунке показаны три фрагмента антенны из первого примера и отмечены расстояния от начала штанги до перекладины.



Чтобы корректно собрать антенну, необходимо соединить в некотором порядке все n фрагментов, при этом расстояние между любыми двумя соседними перекладинами должно быть одинаковым.

На рисунке показан корректный способ соединить фрагменты в первом примере.



К сожалению, члены экспедиции забыли инструкцию по сборке антенны на Земле, а передать её на Марс не представляется возможным — ведь антенна ещё не собрана. Помогите исследователям!

Требуется определить, в каком порядке необходимо соединить фрагменты антенны, чтобы установить связь с Землей.

Входные данные
В первой строке дано одно число n — количество фрагментов ($1 \leq n \leq 100\,000$).

Далее дано описание n фрагментов. В первой строке описания фрагмента даны два целых числа m_i и s_i — количество перекладин и длина штанги в i -м фрагменте ($1 \leq m_i \leq 100\,000$, $0 \leq s_i \leq 10^9$). В следующей строке даны m_i целых чисел $p_{i,j}$ — позиции перекладин, $p_{i,j}$ равно расстоянию в сантиметрах от начала штанги до j -й перекладины на ней ($0 \leq p_{i,1} < p_{i,2} < \dots < p_{i,m_i} \leq s_i$).

Сумма всех m_i не превышает 100 000.

Выходные данные

Если собрать антенну указанным образом возможно, в первой строке выведите «Yes», а во второй строке выведите перестановку чисел от 1 до n — номера фрагментов в порядке, в котором их следует соединить, начало каждого следующего фрагмента в этом порядке присоединяется к концу предыдущего фрагмента. Если существует несколько подходящих ответов, можно вывести любой из них.

Если собрать антенну невозможно, в единственной строке выведите «No».

входные данные
3 1 7 3 1 8 6 2 8 1 6
выходные данные
Yes 2 1 3

входные данные
1 1 7 5
выходные данные
Yes 1

входные данные
1 3 10 2 5 9
выходные данные
No

входные данные
3 1 5 3 1 3 3 1 6 3
выходные данные
No

входные данные
4 1 5 0 1 0 0 1 3 3 1 0 0
выходные данные
Yes 3 2 4 1

04A. Остовное дерево

4 секунды, 256 мегабайт

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

Входные данные

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \leq n \leq 10\,000$). Каждая из следующих n строк содержит два целых числа x_i, y_i — координаты i -й вершины ($-10\,000 \leq x_i, y_i \leq 10\,000$). Никакие две точки не совпадают.

Выходные данные

Первая строка выходного файла должна содержать одно вещественное число — вес минимального остовного дерева.

входные данные
2 0 0 1 1
выходные данные
1.4142135624

04B. Плотное остовное дерево

2 секунды, 256 мегабайт

Требуется найти в графе остовное дерево, в котором разница между весом максимального и минимального ребра минимальна.

Входные данные

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя целыми числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n, 0 \leq |w_i| \leq 10^9$). $n \leq 1000, m \leq 10\,000$.

Выходные данные

Если остовное дерево существует, выведите в первой строке выходного файла YES, а во второй строке одно целое число — минимальную разность между весом максимального и минимального ребра в остовном дереве.

В противном случае в единственной строке выведите NO.

входные данные
4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2
выходные данные
YES 0

04C. Дороги короля

2 секунды, 256 мегабайт

Король Байтазар хочет построить новую сеть дорог в Байтландии, состоящей из n городов. В i -м городе проживает a_i людей.

Изначально в королевстве нет ни одной дороги. Король хочет построить несколько дорог таким образом, чтобы из любого города можно было добраться до любого другого. Также, он хочет минимизировать стоимость постройки этих дорог.

Стоимость дороги, соединяющей города i и j равна $(a_i + a_j) \bmod M$ (если хотите мотивацию такой стоимости, прочитайте английскую версию условия).

Помогите королю Байтазару найти минимальную возможную стоимость постройки такой сети дорог.

Входные данные

На первой строке находятся два целых числа n и M ($1 \leq n \leq 200\,000, 1 \leq M \leq 10^9$).

На второй строке находятся n целых чисел a_i , разделенных пробелом, ($0 \leq a_i < M$).

Выходные данные

Выведите единственное целое число — минимальную стоимость постройки сети дорог.

входные данные
5 9 1 3 5 8 8
выходные данные
6

04D. Fine Study

1 second, 256 megabytes

In College of Fine Study i -th course has Levels from level 0 to level a_i . And at the beginning, Vasya is at Level 0 of every course, and he wants to reach the highest Level of every course.

Fortunately, there are M tutorial classes. The i -th tutoial class requires that students must reach at least Level $L1_i$ of course c_i before class begins. And after finishing the i -th tutorial class, the students will reach Level $L2_i$ of course d_i . The i -th tutoial class costs him m_i .

For example, there is a tutorial class only students who reach at least Level 5 of Math can apply. And after finishing this class, the student's Informatics will reach Level 10 if his Informatics' Level is lower than 10.

Now you task is to help Vasya to compute the minimum cost!

Input

The first line of the input file consists of two positive integers, N ($N \leq 50$) and M ($M \leq 2000$). The following line contains N integers, representing a_1 to a_N . The sum of a_1 to a_N will be not greater than 500. Each of the next M lines contain five integers, indicating $c_i, L1_i, d_i, L2_i$ and m_i ($1 \leq c_i, d_i \leq N, 0 \leq L1_i \leq a_{c_i}, 0 \leq L2_i \leq a_{d_i}, m_i \leq 1000$) for the i -th tutorial class. The courses are numbered from 1 to N .

Output

Output the minimum cost for achieving Vasya's target in a line. If his target can't be achieved, just output -1 .

input
3 4 3 3 1 1 0 2 3 10 2 1 1 2 10 1 2 3 1 10 3 1 1 3 10
output
40

05A. Эвакуация

2 секунды, 256 мегабайт

Одна из Сверхсекретных организаций, чье название мы не имеем право разглашать, представляет собой сеть из N подземных бункеров, соединенных равными по длине туннелями, по которым из любого бункера можно добраться до любого другого (не обязательно напрямую). Связь с внешним миром осуществляется через специальные засекреченные выходы, которые расположены в некоторых из бункеров. Организации понадобилось составить план эвакуации персонала на случай экстренной ситуации. Для этого для каждого из бункеров необходимо узнать, сколько времени потребуется для того, чтобы добраться до ближайшего из выходов. Вам, как специалисту по таким задачам, поручено рассчитать необходимое время для каждого из бункеров по заданному описанию помещения Сверхсекретной организации. Для вашего же удобства бункеры занумерованы числами от 1 до N .

Входные данные

В первой строке записано число N , во второй — число K ($1 \leq N \leq 100\,000$, $1 \leq K \leq N$) — количество бункеров и количество выходов соответственно. Далее через пробел записаны K различных чисел от 1 до N , обозначающих номера бункеров, в которых расположены выходы. Потом идёт целое число M ($1 \leq M \leq 100\,000$) — количество туннелей. Далее вводятся M пар чисел — номера бункеров, соединенных туннелем. По каждому из туннелей можно двигаться в обе стороны. В организации не существует туннелей, ведущих из бункера в самого себя, зато может существовать более одного туннеля между парой бункеров.

Выходные данные

В первой строке выведите N чисел, разделённых пробелом — для каждого из бункеров минимальное время, необходимое чтобы добраться до выхода. Считайте, что время перемещения по одному туннелю равно 1. Во второй строке выведите N чисел — для каждого бункера номер ближайшего бункера с выходом, если таких несколько выведите бункер с наименьшим номером.

входные данные
3 1 2 3 1 2 3 1 2 3
выходные данные
1 0 1 2 2 2

05B. Roadblock

2 секунды, 256 мегабайт

Каждое утро Фермер Джон идет от дома к амбару. Ферма представляет собой множество из n полей (дом на поле 1, амбар на поле n), соединенных m двусторонними дорогами, с каждой из которых ассоциирована длина.

Никакие два поля не соединены более чем одной дорогой, и существует маршрут дорог от любого поля к любому. Когда ФД идет от одного поля к другому, он всегда выбирает маршрут, состоящий из последовательности дорог, которые дают минимальную суммарную длину.

Коровы решили сделать ФД маленькую неприятность, выложив сено на одной из m дорог, тем самым удваивая ее длину.

Коровы хотят выбрать такую дорожку, чтобы максимально увеличить расстояние, которое ФД пройдет от дома до амбара. Помогите коровам определить, насколько они удлинят маршрут ФД.

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n \leq 300$; $1 \leq m \leq 90\,000$).

Следующие m строк описывают дороги. i -я из этих строк состоит из трех целых чисел u_i , v_i и l_i — дорога длины l_i соединяет поля с номерами v_i и u_i ($1 \leq u_i, v_i \leq n$; $1 \leq l_i \leq 10^6$).

Выходные данные

Выведите максимально возможное увеличение общей длины кратчайшего маршрута, которого можно добиться удвоением длины одной дороги.

входные данные
5 7 2 1 5 1 3 1 3 2 8 3 5 7 3 4 3 2 4 7 4 5 2
выходные данные
2

Если коровы удвоят длину дороги от поля 3 к полю 4 (от 3 до 6), тогда кратчайшим маршрутом станет путь 1-3-5, с общей длиной $1 + 7 = 8$. Что на 2 больше, чем исходный кратчайший маршрут.

05C. Гениальная прогулка

2 секунды, 512 мегабайт

В новом регионе Сэм обнаружил n городов, соединенных m двусторонними дорогами. Сэм может перемещаться только по дорогам. Ему нужно добраться из города s в город t , и при этом не попасть под темпоральный дождь. Согласно прогнозу погоды, дождь над i -й дорогой будет идти в отрезки времени $[(a_i + b_i) \cdot k + a_i, (a_i + b_i) \cdot (k + 1)]$ для всех целых k (a_i и b_i — положительные). Чтобы пройти по i -й дороге, Сэм должен потратить d_i времени, и на протяжении всего этого времени над этой дорогой не должен идти дождь. В городах Сэм может укрыться от дождя, поэтому в них он может находиться в любое время. Также, Сэм может выйти из города на дорогу в момент окончания дождя и зайти в город с дороги в момент начала дождя.

В момент времени 0, Сэм находится в городе s , и интересуется, в какой минимальный момент времени он может оказаться в городе t . Помогите ему ответить на этот вопрос.

Входные данные

В первой строке даны четыре целых числа n, m, s и t — количество городов, дорог, стартовый и конечный город соответственно ($1 \leq n \leq 100\,000$; $0 \leq m \leq 200\,000$; $1 \leq s, t \leq n$). В следующих m строках дано описание дорог. В каждой строке дано пять целых чисел u_i, v_i, a_i, b_i и d_i ($1 \leq u_i, v_i \leq n$; $1 \leq a_i, b_i, d_i \leq 10^9$). Дорога номер i соединяет города u_i и v_i .

Выходные данные

Если Сэм не может добраться из города s до города t , выведите «−1», иначе выведите минимальный момент времени, в который он может оказаться в городе t .

входные данные
3 2 1 3 1 2 3 4 1 2 3 2 3 2
выходные данные
7

06A. Pink Floyd

1 секунда, 256 мегабайт

Группа Pink Floyd собирается дать новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист Роджер Уотерс постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным. Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

Входные данные

Первая строка входного файла содержит три натуральных числа n , m и k — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ($n \leq 100, m \leq 10^4, 2 \leq k \leq 10^4$). Города пронумерованы числами от 1 до n . Следующие m строк содержат описание рейсов, по одному на строке. Рейс номер i описывается тремя числами b_i, e_i и w_i — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ($1 \leq b_i, e_i \leq n, -10^5 \leq w_i \leq 10^5$). Последняя строка содержит числа a_1, a_2, \dots, a_k — номера городов, в которых проводятся концерты. В начале концертного тура группа находится в городе a_1 . Гарантируется, что группа может дать все концерты.

Выходные данные

Первая строка выходного файла должна содержать число s — количество рейсов, которые должна сделать группа. Вторая строка должна содержать s чисел — номера используемых рейсов. Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку «infinitely kind».

входные данные
4 8 5 1 2 -2 2 3 3 3 4 -5 4 1 3 1 3 2 3 1 -2 3 2 -3 2 4 -10 1 3 1 2 4
выходные данные
6 5 6 5 7 2 3

входные данные
4 8 5 1 2 -2 2 3 3 3 4 -5 4 1 3 1 3 2 3 1 -2 3 2 -3 2 4 10 1 3 1 2 4
выходные данные
infinitely kind

06B. Егор и граф

3 секунды, 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i -том шаге Егор удаляет из графа вершину номер x_i . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i , то Егор хочет знать значение следующей суммы: $\sum_{v, u, v \neq u} d(i, v, u)$.

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Входные данные

В первой строке содержится целое число n ($1 \leq n \leq 500$) — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j -тое число в i -той строке a_{ij} ($1 \leq a_{ij} \leq 10^5, a_{ii} = 0$) обозначает вес ребра, ведущего из вершины i в вершину j .

В следующей строке содержится n различных целых чисел: x_1, x_2, \dots, x_n ($1 \leq x_i \leq n$) — вершины, которые удаляет Егор.

Выходные данные

Выведите n целых чисел — i -тое число равно искомой сумме перед i -тым шагом.

входные данные
2 0 5 4 0 1 2
выходные данные
9 0

07A. Кибер-взлом

2 s., 512 MB

Ви пытается взломать сервера корпорации Арасака, чтобы отключить охрану и проникнуть в их офис. Искусственный интеллект сервера пытается ему в этом помешать.

Взлом происходит следующим образом. Рассмотрим ориентированный граф, на каждом ребре которого написана буква английского алфавита. Граф может содержать кратные ребра и даже петли. У Ви есть токен, изначально находящийся в некоторой вершине v , и у ИИ сервера есть токен, изначально находящийся в некоторой вершине u . Затем они по-очереди совершают ходы, Ви ходит первым. На своём ходу Ви выбирает произвольное ребро, исходящее из вершины, в которой находится его токен. Он перемещает токен по этому ребру, а также пытается произвести атаку типа c , где c — символ, написанный на выбранном ребре. ИИ на своём ходу также выбирает одно из рёбер, исходящих из вершины, в которой находится его токен, и перемещает токен по этому ребру. При этом, чтобы успешно отразить атаку, он должен выбрать ребро, на котором написан тот же символ c .

Если Ви не может сделать ход, потому что из вершины, в которой находится его токен, не исходит ни одного ребра, взлом завершается провалом. Если ИИ не может выбрать ребро, исходящее из вершины, в которой находится его токен, на котором написан символ c , взлом завершается успешно. Также, возможна ситуация, в которой Ви и ИИ будут делать ходы бесконечно долго.

Помогите Ви определить количество стартовых состояний, то есть пар вершин v и u , при которых взлом будет произведен успешно при оптимальных действиях Ви и ИИ.

Входные данные

В первой строке даны два целых числа n и m — количество вершин и ребер в графе ($1 \leq n \leq 1\,000, 0 \leq m \leq 1\,000$).

В следующих m строках дано описание ребер графа. Каждая строка содержит два целых числа a_i и b_i и строчную букву английского алфавита c_i , они обозначают ребро из вершины a_i в вершину b_i , на котором написан символ c_i ($1 \leq a_i, b_i \leq n$).

Выходные данные

Выведите одно число — искомое количество стартовых состояний.

входные данные
3 3 1 2 a 2 3 b 3 1 c
выходные данные
6

входные данные
5 10 2 2 c 3 5 b 5 4 b 2 3 b 3 5 c 3 1 b 4 2 a 4 4 a 2 4 b 2 5 c
выходные данные
15

В первом примере, если изначально токены Ви и ИИ стоят в одной и той же вершине, процесс никогда не завершится. Во всех остальных случаях, взлом будет успешным.

07B. Жестокая задача

2 секунды, 256 мегабайт

Штирлиц и Мюллер стреляют по очереди. В очереди n человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

Входные данные

Входной файл содержит единственное число n ($2 \leq n \leq 5\,000$) — количество человек в очереди.

Выходные данные

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова `Mueller`. Иначе в первой строке необходимо вывести слово `Schtirlitz`, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

входные данные
3
выходные данные
Schtirlitz 2

входные данные
4
выходные данные
Mueller