Задача A. Лабиринт

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В одном из уровней компьютерной игры вы попали в лабиринт, состоящий из n строк, каждая из которых содержит m клеток. Каждая клетка либо свободна, либо занята препятствием. Стартовая клетка находится в строке r и столбце c. За один шаг вы можете переместиться на одну клетку вверх, влево, вниз или вправо, если она не занята препятствием. Вы не можете перемещаться за границы лабиринта.

К сожалению, ваша клавиатура крайне близка к поломке, поэтому вы можете переместиться влево не более x раз и вправо не более y раз. При этом ограничений на перемещения вверх и вниз нет, поскольку клавиши, используемые для движения вверх и вниз, всё ещё в идеальном состоянии.

Теперь вы для каждой клетки поля решили установить, можно ли выбрать такую последовательность нажатий, которая приведёт вас из стартовой в эту клетку. Посчитайте, сколько клеток поля обладают таким свойством.

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leqslant n, m \leqslant 2000$) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит два целых числа r, c ($1 \leqslant r \leqslant n, 1 \leqslant c \leqslant m$) — номер строки и столбца, на пересечении которых расположена стартовая клетка.

Третья строка содержит два целых числа x, y ($0 \leqslant x, y \leqslant 10^9$) — максимальное количество перемещений влево и вправо, соответственно.

Следующие n строк содержат описание лабиринта. Каждая из этих строк имеет длину m и состоит только из символов '.' и '*'. В i-й строке j-й символ соответствует клетке лабиринта с номерами строки и столбца i и j, соответственно. Символ '.' соответствует свободной клетке лабиринта, а символ '*' — клетке с препятствием.

Гарантируется, что стартовая клетка не занята препятствием.

Формат выходных данных

Выведите одно число — количество клеток лабиринта, достижимых из стартовой, включая её саму.

Примеры

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>4 5
3 2
1 2
.....
.***.
...**
*....</pre></td><td><pre>10</pre></td></tr>
<tr><td><pre>5 5
5 4
3 1
**...
**.*.
...*.
.***.
.....</pre></td><td><pre>16</pre></td></tr>
</table>

Задача B. RMQ

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Есть массив из N целых чисел и M запросов вида: найдите минимум на отрезке с концами l_i, r_i.

Формат входных данных

Входной файл содержит T наборов тестовых данных. Каждый набор тестовых данных задаётся числами N, M, A, B ($1 \leqslant N \leqslant 25\,000, 1 \leqslant A, B \leqslant 10^9$), где N — размер массива, M — число запросов.

Массив и запросы нужно получить следующим образом: выпишем последовательность чисел $C_i = (A \cdot i + B) \bmod 2^{32}$.

Элементы последовательности с номерами от 1 до N — элементы массива. Элементы последовательности с номерами от $N + 1$ до $N + 2 \cdot M$ взятые по модулю N образуют M пар чисел, которые являются границами отрезков запросов. Ввод заканчивается числами 0 0 0 0. Массив индексируется с нуля.

Сумма N по всем наборам тестовых данных не превосходит 10^8. Сумма M по всем наборам тестовых данных не превосходит $2 \cdot 10^7$.

Формат выходных данных

Для каждого набора тестовых данных выведите сумму по всем запросам.

Пример

стандартный ввод	стандартный вывод
`10 10 955379886 619166003` `0 0 0 0`	`7671393960`

Замечание

Массив:

```
1574545889 2529925775 3485305661 145718251 1101098137 2056478023 3011857909
3967237795 627650385 1583030271
```

Запросы:

```
7 3
3 9
5 1
7 7
3 9
5 5
1 7
3 9
9 5
1 7
```

Задача C. Шарады

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	64 мегабайта

В ещё не изведанной части вселенной есть планета, на которой живут одни математики. На этой планете живут N математиков, каждый — в своём городе. Никакие два города не соединены дорогами, потому что математики могут общаться онлайн, оставляя комментарии о научных трудах друг друга.

Всё шло тихо и спокойно, пока один математик не решил написать научную работу со своего мобильного телефона. Автоисправление в телефоне заменило «очевидно» на «шарада». Не перечитав свою работу, математик так и опубликовал её. Совсем скоро об игре в шарады узнали все математики планеты, и им захотелось собраться и поиграть всем вместе. Поэтому в скором времени началась постройка дорог между городами. Строительство дорог будет идти M дней в соответствии со следующим расписанием: в первый день строятся дороги между всеми парами городов, у номеров которых наибольший общий делитель равен M. Во второй день строятся дороги между всеми парами городов, наибольший делитель номеров которых равен $M - 1$. И так далее до M-го дня, в который дороги строятся между всеми парами городов с взаимно простыми номерами. Говоря более формально, в i-й день (нумеруя дни с единицы) дороги строятся между всеми такими парами городов A и B, что $\text{НОД}(A, B) = M + 1 - i$.

Математики очень заняты постройкой дорог, поэтому они просят вас помочь определить минимальное число дней с начала строительства, через которое данная пара математиков сможет встретиться, чтобы поиграть в шарады.

Формат входных данных

В первой строке даны три целых целых положительных числа N, M и Q ($1 \leqslant N, Q \leqslant 100\,000$, $1 \leqslant M \leqslant N$) - количество городов, длительность строительства дорог и количество запросов соответственно.

В следующих Q строках вводятся по два целых числа A и B ($1 \leqslant A, B \leqslant N$) — номера городов двух математиков, которым интересно, через сколько дней они смогут встретиться (добраться из одного город в другой, проехав по уже построенным дорогам).

Формат выходных данных

На каждый из Q запросов выведите ответы — Q чисел, каждое в отдельной строке.

Система оценки

Программы, правильно работающие при $N \leqslant 1000$, будут оцениваться в 40 баллов.

Примеры

стандартный ввод	стандартный вывод
8 3 3 2 5 3 6 4 8	3 1 2
25 6 1 20 9	4
9999 2222 2 1025 2405 3154 8949	1980 2160

Замечание

Пояснение к первому тесту:

В первый день строится дорога $(3, 6)$. Поэтому ответ на второй запрос 1. На второй день строятся дороги $(2, 4)$, $(2, 6)$, $(2, 8)$, $(4, 6)$ и $(6, 8)$. Города 4 и 8 теперь связаны (можно добраться из первого во второй используя город 6). На третий день строятся дороги между взаимно простыми городами, поэтому города 2 и 5 оказываются соединены.

Пояснение ко второму тесту:

На второй день строится дорога $(20, 15)$, на четвертый день — дорога $(15, 9)$. Таким образом, начиная с четвёртого дня, города, 20 и 9 связаны (через город 15).

Задача D. Дед и мопед

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Дед Максим собирается в путешествие по Флатландии. К сожалению, из доступных средств передвижения у него есть только мопед, запас хода которого ограничен. Более точно, если бак мопеда полностью заполнен, то мопед может проехать не более s километров без дополнительной дозаправки.

Всего во Флатландии есть n городов, пронумерованных от 1 до n. В некоторых городах находятся заправки, и если в городе есть заправка, то в этом городе дед Максим может полностью наполнить бак. К сожалению, заправки присутствуют лишь в k городах. Также во Флатландии есть m дорог, i-я из которых соединяет города u_i и v_i и имеет длину c_i километров. По каждой дороге можно перемещаться в обоих направлениях.

Дед Максим начинает свое путешествие в городе с номером 1 с полным баком (в городе 1 есть заправка). Помогите ему определить, до каких городов он сможет добраться на мопеде.

Формат входных данных

В первой строке записаны четыре целых числа n, m, k, s ($1 \leqslant n \leqslant 150\,000$, $0 \leqslant m \leqslant 150\,000$, $1 \leqslant k \leqslant n$, $1 \leqslant s \leqslant 10^9$) — количество городов, количество дорог, количество заправок и объем бака.

В следующих m строках записаны по три целых числа u_i, v_i, c_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$, $1 \leqslant c_i \leqslant s$) — начало, конец и длина i-й дороги. Гарантируется, что не существует двух дорог, соединяющих одинаковую пару городов.

В следующей строке записаны k целых чисел p_i ($1 \leqslant p_i \leqslant n$) — номера городов с заправками. Гарантируется, что заправка присутствует в городе 1.

Формат выходных данных

В первой строке выведите единственное число x: количество гордов, до которых дед Максим может добраться.

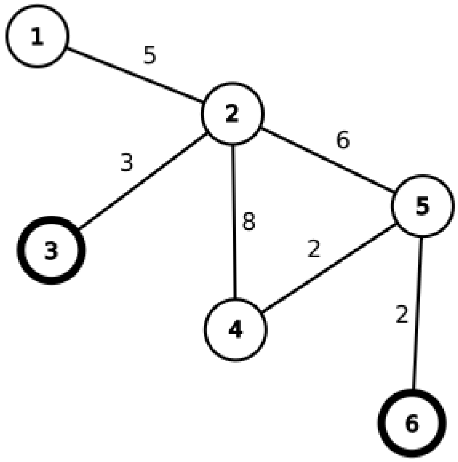
Во второй строке выведите x целых чисел — номера подходящих городов **в порядке возрастания**.

Пример

стандартный ввод	стандартный вывод
6 6 3 10 1 2 5 2 3 3 2 4 8 2 5 6 4 5 2 5 6 2 1 3 6	4 1 2 3 5

Замечание

Рисунок ниже иллюстрирует первый пример из условия:



Из начального города с номером 1 можно доехать до городов с номерами 2 и 3 без дополнительных дозаправок. Также можно доехать до города с номером 3, пополнить там бак и доехать до города с номером 5. До городов 4 и 6 добраться невозможно, так как минимальное расстояние от достижимой заправки до них равно 11.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		k	n	c_i		
0	0	–	–	–	–	Тесты из условия.
1	15	–	$n \leqslant 100$	–	0	
2	11	$k = 1$	$n \leqslant 5000$	–	–	
3	8	$k = 1$	–	–	2	
4	12	–	$n \leqslant 5000$	$c_i = 1$	–	
5	9	–	$n \leqslant 5000$	$c_i \leqslant 10$	0, 4	
6	17	–	$n \leqslant 5000$	–	0, 1, 2, 4, 5	
7	28	–	–	–	0–6	**Offline-проверка.**

Задача E. Декартово

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Государство Иксово состоит из N_x городов, некоторые пары которых связаны дорогами с двусторонним движением. Каждая дорога имеет свою длину. Всего межгородских дорог в стране M_x, причем известно, что из каждого города Иксевщины можно доехать по дорогам до каждого другого города этой страны. Города Иксово пронумерованы натуральными числами от 1 до N_x.

Государство Игреково состоит из N_y городов, некоторые пары которых связаны дорогами с двусторонним движением. Каждая дорога имеет свою длину. Всего межгородских дорог в стране M_y, причем известно, что из каждого города Игреково можно доехать по дорогам до каждого другого города этой страны. Города Игреково пронуменрованы натуральными числами от 1 до N_y.

Страна Декартово состоит из $N = N_x \cdot N_y$ городов: каждому городу Декартово во взаимно однозначное соответствие можно поставить пару городов-побратимов (x, y) , где x — город Иксово, а y — город Игреково. Некоторые пары городов Декартово также соединены дорогами с двусторонним движением. Дорог в стране ровно $M = N_x \cdot M_y + N_y \cdot M_x$. При этом дорога между городами (x_1, y_1) и (x_2, y_2) существует только в одном из таких двух случаев:

1. Если $x_1 = x_2$, а между городами y_1 и y_2 Игреково проложена дорога. При этом длина дороги между городами (x, y_1) и (x, y_2) Декартово равно длине дороги между городами y_1 и y_2 Игреково.

2. Если $y_1 = y_2$, а между городами x_1 и x_2 Иксевщены проложена дорога. При этом длина дороги между городами (x_1, y) и (x_2, y) Декартово равно длине дороги между городами x_1 и x_2 Иксево.

Города разных государств между собой дорогами не соединены.

Данная задача состоит из двух подзадач. В обеих подзадачах всю информацию про соединение дорогами задано во входных файлах.

В первой подзадаче требуется определить длину самого короткого пути по дорогам Декартовщины из города $(1, 1)$ в город (N_x, N_y).

Во второй подзадаче некоторые дороги Декартовщины требуется закрыть. Ваша задача — определить, дороги какой наименьшей суммарной длины можно оставить в Декартовщине, чтобы из любого ее города все еще можно было попасть в любой другой.

Формат входных данных

Первая строка входного файла содержит номер подзадачи, которую требуется решить (1 или 2). Вторая строка содержит натуральные числа N_x и M_x $(1 \leqslant N_x, M_x \leqslant 5 \cdot 10^4)$ — количество городов и дорог в Иксово. В последующих M_x строках описаны дороги Иксово: в каждой строке по три числа, где первые два задают номера разных городов, соедененных дорогой, а третья есть длиной соответствующей дороги (натуральное число, которое не превышает 10^7).

В следующей строке входного файла указаны натуральные числа N_y и M_y $(1 \leqslant N_y, M_y \leqslant 5 \cdot 10^4)$ — количество городов и дорог в Игреково. Последующие M_y строк содержат описание дорог Игреково; формат данных и ограничения соответствуют описанным выше.

Формат выходных данных

Выведите единственное целое число — ответ на вопрос подзадачи.

Примеры

стандартный ввод	стандартный вывод
1 3 2 2 1 15 3 1 14 3 2 2 1 15 3 2 15	44
2 3 2 2 1 15 3 1 14 3 2 2 1 15 3 2 15	117

Задача F. Путешествующий торговец

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вы прибыли в Австралию, где есть n рынков, соединённых m односторонними дорогами, путешествие по каждой дороге занимает определённое количество минут.

На рынках торгуются k предметами. Каждый предмет имеет определённую стоимость покупки или продажи. Бывает так, что на рынке можно только купить товар или только продать товар, а также бывает, что рынку вообще не интересен товар. Вы можете считать, что если на рынке есть товар, его есть бесконечно много, а также, если рынок готов покупать товар, он готов покупать бесконечно много.

Чтобы как можно быстрее заработать денег вы хотите найти самый эффективный цикл. Цикл — это путь, который начинается в каком-то рынке v с пустым рюкзаком, проходит по дорогам и рынкам (возможно, по пути покупаются и продаются товары), и возвращается в вершину v, опять с пустым рюкзаком. Цикл может посещать дорогу или рынок несколько раз. Когда вы покупаете товар, вы кладёте его в рюкзак. Однако в рюкзак можно положить **не более одного товара**. Вы можете считать, что независимо от того, сколько у вас денег, вы можете купить товар.

Выгода цикла — это суммарное количество денег, которое вы заработали на продажах, минус количество денег, которые вы потратили на покупку. Длительность цикла — количество минут, которые вы потратите, чтобы пройти его. Эффективность цикла — отношение его выгоды к длительности.

Найдите максимальную эффективность среди всех циклов со строго положительной длительностью. Вы должны найти это значение, округленное вниз. Если такого цикла не существует, ответ равен 0.

Формат входных данных

Первая строка содержит три целых числа n, m, k ($1 \leqslant n \leqslant 100$, $1 \leqslant m \leqslant 9900$, $1 \leqslant k \leqslant 1000$).

Затем следуют n строк, i-я из которых содержит $2k$ чисел $b_{i,1}, s_{i,1}, b_{i,2}, s_{i,2}, \ldots, b_{i,k}, s_{i,k}$ ($0 < s_{i,j} \leqslant b_{i,j} \leqslant 10^9$). Для всех $1 \leqslant j \leqslant k$ пара чисел $b_{i,j}$ и $s_{i,j}$ означает цену, по которой вы можете купить и продать товар j на i-м рынке, соответственно. Если товар не может быть куплен или продан, тогда значение равно -1.

Далее следуют m строк, p-я из которых содержит три целых числа v_p, w_p и t_p ($v_p \neq w_p$, $1 \leqslant t_p \leqslant 10^7$), описывающих дорогу из v_p в w_p, которая занимает t_p минут.

Гарантируется, что не существует такой пары рёбер $1 \leqslant p < q \leqslant m$, что $(v_p, w_p) = (v_q, w_q)$.

Формат выходных данных

Выведите одно число — ответ.

Пример

стандартный ввод	стандартный вывод
4 5 2 10 9 5 2 6 4 20 15 9 7 10 9 -1 -1 16 11 1 2 3 2 3 3 1 4 1 4 3 1 3 1 1	2

Задача G. Разностный MST

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дан массив $x_1, x_2, \ldots, x_n$.

Давайте создадим неориентированный граф на n вершинах, в котором изначально нет ребер.

После этого для каждой пары (u, v), такой что $u < v$, добавим в граф ребро между вершинами u и v веса $x_v - x_u$.

Найдите вес минимального остовного дерева в получившемся графе.

Формат входных данных

Первая строка входных данных содержит одно целое положительное число t $(1 \leqslant t \leqslant 300\,000)$ — количество тестовых примеров.

Первая строка каждого тестового примера содержит одно целое положительное число n $(1 \leqslant n \leqslant 300\,000)$ — количество элементов массива.

Вторая строка каждого тестового примера содержит n целых чисел $x_1, x_2, \ldots, x_n$ $(-300\,000 \leqslant x_i \leqslant 300\,000)$ — элементы массива.

Гарантируется, что сумма n по всем тестовым примерам не превышает $300\,000$.

Формат выходных данных

Для каждого тестового примера выведите одно целое число — вес минимального остовного дерева в данном графе.

Пример

стандартный ввод	стандартный вывод
2 5 1 2 3 4 5 3 10 45 10	4 -35

Задача H. Химическая таблица

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Учёные Иннополиса продолжили исследование периодической таблицы. Существуют $n \cdot m$ известных элементов, и они представлены в периодической таблице — прямоугольнике, состоящем из n строк и m столбцов. Каждый элемент может быть описан своими координатами в таблице (r, c) $(1 \leqslant r \leqslant n,\ 1 \leqslant c \leqslant m)$.

Недавно учёные открыли, что для каждых четырёх различных элементов в этой таблице, которые образуют прямоугольник со сторонами, параллельными сторонам таблицы, если они имеют экземпляры трёх из четырёх элементов, то с помощью ядерного синтеза они могут произвести четвёртый элемент. Так, если имеются элементы с позиций (r_1, c_1), (r_1, c_2), (r_2, c_1), где $r_1 \neq r_2$ и $c_1 \neq c_2$, то можно произвести элемент (r_2, c_2).

Использованные экземпляры элементов не выбрасываются и могут быть использованы в дальнейшем для создания других элементов. Созданные элементы также могут в этом участвовать.

Учёные Иннополиса уже имеют образцы q элементов. Они хотят получить образцы всех $n \cdot m$ элементов таблицы. Чтобы добиться этого, они могут купить некоторые образцы в других лабораториях, а потом произвести остальные в некотором порядке. Помогите им определить, какое минимальное число элементов им надо купить.

Формат входных данных

Первая строка входных данных содержит три целых числа n, m, q $(1 \leqslant n, m \leqslant 200\,000;$ $0 \leqslant q \leqslant \min(n \cdot m, 200\,000))$ — размеры таблицы элементов и количество элементов, которые учёные уже имеют.

Следующие q строк содержат по два целых числа каждая: r_i, c_i $(1 \leqslant r_i \leqslant n,\ 1 \leqslant c_i \leqslant m)$, которые описывают расположение уже имеющихся элементов в таблице.

Формат выходных данных

Выведите минимальное количество элементов, которые надо купить.

Примеры

стандартный ввод	стандартный вывод
2 2 3 1 2 2 2 2 1	0
1 5 3 1 3 1 1 1 5	2
4 3 6 1 2 1 3 2 2 2 3 3 1 3 3	1

Задача I. Проездной

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

JOI-кун живёт в городе с n станциями, занумерованными от 1 до n. Есть m железных дорогов, пронумерованных от 1 до m, и i-я дорога соединяет станции a_i и b_i в обоих направлениях, притом стоимость проезда равна c_i.

JOI-кун живёт рядом со станцией s и ходит в старшую школу IOI рядом со станцией t. Он планирует купить проездной между этими станциями. Когда он покупает проездной, он должен выбрать какой-то кратчайший путь между s и t. Используя проездной, он может использовать любую железную дорогу на выбранном пути в обоих направлениях, не платя за проезд.

JOI-кун часто ездит в книжные маганизы рядом со станциями u и v. Поэтому он хочет купить проездной так, чтобы минимальная стоимость проезда от u до v была минимальна.

Когда он перемещается из станции u на станцию v, он сперва выбирает путь от станции u до станции v. Тогда за дороги в пути, входящие в проездной, он заплатит 0 иен, а за не входящие в проездной — стоимость проезда через них, то есть c_i для дороги i.

Найдите минимально возможную стоимость пути из u в v, если JOI-кун выберет путь для проездного оптимально.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 2 \cdot 10^5$).

Вторая строка содержит два целых числа s и t ($1 \leqslant s, t \leqslant n$, $s \neq t$).

Третья строка содержит два целых числа u и v ($1 \leqslant u, v \leqslant n$, $u \neq v$, $s \neq u$ или $t \neq v$).

Далее следуют m строк, описывающие железные дороги, i-я из них содержит три целых числа a_i, b_i, c_i ($1 \leqslant a_i < b_i \leqslant n$, $1 \leqslant c_i \leqslant 10^9$). Для всех $1 \leqslant i < j \leqslant m$ верно $a_i \neq a_j$ или $b_i \neq b_j$.

Гарантируется, что JOI-кун может добраться от любой станции до любой другой, перемещаясь только по железным дорогам.

Формат выходных данных

Выведите одно число — ответ.

Примеры

стандартный ввод	стандартный вывод
6 6 1 6 1 4 1 2 1 2 3 1 3 5 1 2 4 3 4 5 2 5 6 1	2
6 5 1 2 3 6 1 2 1000000000 2 3 1000000000 3 4 1000000000 4 5 1000000000 5 6 1000000000	3000000000
8 8 5 7 6 8 1 2 2 2 3 3 3 4 4 1 4 1 1 5 5 2 6 6 3 7 7 4 8 8	15
5 5 1 5 2 3 1 2 1 2 3 10 2 4 10 3 5 10 4 5 10	0

Задача J. Трафик

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Центр Гдыни находится на острове по середине реки Кача. Каждое утро тысячи машин проезжают через этот остров из спальных районов на западном берегу реки к индустриальным районам на восточном берегу.

Остров представляет из себя прямоугольник, стороны которого параллельны осям координат. Представим его, как прямоугольник $A \times B$, противоположные углы которого находятся в точках $(0, 0)$ и (A, B). На острове есть n перекрестков, пронумерованных от 1 до n. Перекресток с номером i находится в точке (x_i, y_i). Если перекресток находится в точках вида $(0, y)$, он находится на западной части острова, а если в точках вида (A, y) — на восточной. Перекрестки соединены улицами. Каждая улица — это отрезок, соединяющий два перекрестка. Улицы бывают как односторонние, так и двусторонние. Никакие улицы не пересекаются, кроме как в перекрестках, которые являются концами улиц. На острове нет мостов и туннелей. Из-за растущей загруженности дорог мэр города нанял вас проверить, сколько перекрестков в восточной части острова достижимы по улицам из каждого перекрестка западной части.

Формат входных данных

Первая строка содержит четыре целых числа n, m, A и B ($1 \leqslant n \leqslant 3 \cdot 10^5$, $0 \leqslant m \leqslant 9 \cdot 10^5$, $1 \leqslant A, B \leqslant 10^9$) — число перекрестков, число улиц и размеры острова, соответственно.

В каждой из следующих n строк содержится по два целых числа x_i, y_i ($0 \leqslant x_i \leqslant A$, $0 \leqslant y_i \leqslant B$), которые описывают координаты перекрестка i. Никакие два перекрестка не находятся в одной точке.

Следующие m строк описывают улицы. Каждая улица описывается тремя целыми числами c_i, d_i, k_i ($1 \leqslant c_i, d_i \leqslant n$, $c_i \neq d_i$, $k_i \in \{1, 2\}$), описывающими улицу, соединяющую перекрестки c_i и d_i. Причем, если $k_i = 1$, то улица односторонняя из c_i в d_i, а иначе по улице можно ездить в обоих направлениях. Каждая упорядоченная пара (c_i, d_i) встречается не более одного раза.

Гарантируется, что существует хотя бы один перекресток в западной части, из которого можно добраться до какого-нибудь перекрестка восточной части острова.

Формат выходных данных

Выведите по одной строке для каждого перекрестка на западной стороне острова, в порядке убывания y-координаты перекрестка.

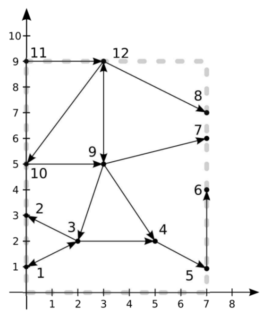
Строка должна содержать число перекрестков левой стороны острова, достижимых из соответствующего перекрестка западной стороны.

Примеры

стандартный ввод	стандартный вывод
5 3 1 3 0 0 0 1 0 2 1 0 1 1 1 4 1 1 5 2 3 5 2	2 0 2
12 13 7 9 0 1 0 3 2 2 5 2 7 1 7 4 7 6 7 7 3 5 0 5 0 9 3 9 1 3 2 3 2 1 3 4 1 4 5 1 5 6 1 9 3 1 9 4 1 9 7 1 9 12 2 10 9 1 11 12 1 12 8 1 12 10 1	4 4 0 2

Замечание

Картинка ко второму примеру



Задача K. Кукушки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Британские учёные решили заняться орнитологией и понаблюдать за жизнью необычных кукушек. Для этого они вырастили дерево и построили на нём n гнёзд, в каждом из которых живёт кукушка. Наблюдение за деревом состоит в том, что в некоторые моменты времени учёные оценивают, можно ли подложить определённое яйцо в гнездо к некоторой кукушке или нет.

Каждое яйцо может вынашиваться только в двух определённых гнёздах. Каждое яйцо задаётся неупорядоченной парой различных чисел (x, y). Яйцо (x, y) может вынашиваться в любом из гнёзд x и y и не может вынашиваться в других гнёздах. Обратите внимание, яйцо (x, y) не отличается от яйца (y, x).

Теперь опишем процесс подкладывания яйца в имеющиеся гнезда: пусть учёные хотят подложить яйцо (x, y) в гнездо x. Если в гнезде x нет яйца, то яйцо (x, y) просто остаётся в этом гнезде, и процесс на данном шаге завершается. Если же в гнезде x лежит какое-то яйцо (x, p), то кукушка кладёт яйцо (x, y) в данное гнездо, а яйцо (x, p) пытается подложить в гнездо p аналогичным образом, и процесс продолжается.

Вам предлагается отвечать на вопросы учёных. Всего есть три типа вопросов:

1. (Теоретический) Закончится ли процесс, если подложить яйцо (x, y) в гнездо x? Так как вопрос чисто теоретический, оно **не добавляется** на самом деле, и состояние гнёзд не меняется.

2. (Практический) Закончится ли процесс, если подложить яйцо (x, y) в гнездо x? Если процесс закончится, то яйцо **добавляется** в реальности согласно описанному процессу.

3. (Теоретический) Сколько существует **упорядоченных** пар различных чисел (x, y), таких что яйцо (x, y) можно подложить в гнездо x с учётом имеющихся в гнёздах яиц? При этом для каждого яйца ответ определяется независимо от других добавляемых яиц.

Формат входных данных

В первой строке вводятся три целых числа n, m, q, ($2 \leqslant n \leqslant 200\,000$, $0 \leqslant m \leqslant n$, $1 \leqslant q \leqslant 600\,000$), где n — количество гнёзд на дереве, m —количество яиц, которые учёные уже положили, q — количество вопросов, которые задают учёные.

В каждой из m последующих строк следуют по два числа x_i, y_i, означающих, что в гнезде x_i лежит яйцо (x_i, y_i). Гарантируется, что все x_i различны и что $x_i \neq y_i$ для всех i.

В следующих q строках описаны вопросы учёных. Вопросы даны в том порядке, в котором на них требуется отвечать. Первое число t_j в строке описывает тип вопроса.

Если $t_j = 1$ или $t_j = 2$, то далее идут два различных числа x_j и y_j, описывающих яйцо, которое фигурирует в соответствующем вопросе.

Если $t_j = 1$, то яйцо не требуется добавлять в текущую расстановку.

Если $t_j = 2$, то яйцо требуется добавить, если процесс добавления потребует конечного числа перекладываний.

Если $t_j = 3$, то требуется определить количество упорядоченных пар (x, y), таких что яйцо (x, y) можно добавить в гнездо x с тем, чтобы процесс когда-нибудь завершился. В реальности никакие яйца в расстановку не добавляются.

Формат выходных данных

Для каждого вопроса первого и второго типа выведите единственное слово «`Yes`» или «`No`» в зависимости от того, закончится ли процесс перекладывания.

Для каждого запроса третьего типа выведите количество искомых упорядоченных пар.

Система оценки

Тесты к этой задаче
прохождении всех тестов
Пусть t_1 — количество з
чество запросов третьего
данной группе станут до

Группа	Баллы	Дополнительные ограничения				Необх. группы	Комме
		n	t_1	t_2	t_3		
0	0	–	–	–	–	–	При
1	13	$n \leqslant 2000$	$t_1 \leqslant 2000$	$t_2 = 0$	$t_3 = 0$	–	
2	14	$n \leqslant 2000$	$t_1 \leqslant 2000$	$t_2 = 0$	$t_3 \leqslant 1$	1	
3	12	$n \leqslant 2000$	$t_1 \leqslant 2000$	$t_2 \leqslant 2000$	$t_3 \leqslant 2000$	0 – 2	
4	12	–	$t_1 \leqslant 2 \cdot 10^5$	$t_2 = 0$	$t_3 = 0$	1	
5	18	–	$t_1 \leqslant 2 \cdot 10^5$	$t_2 = 0$	$t_3 \leqslant 1$	1 – 2, 4	
6	31	–	$t_1 \leqslant 2 \cdot 10^5$	$t_2 \leqslant 2 \cdot 10^5$	$t_3 \leqslant 2 \cdot 10^5$	0 – 5	**Offline-п**

Пример

стандартный ввод	стандартный вывод
5 3 8 1 2 5 1 2 4 1 1 2 3 2 1 2 3 2 4 2 2 5 3 3 1 4 5	Yes 20 Yes 8 No Yes 0 No

Замечание

Изначальное расположение яиц в тесте из условия такое: в первом гнезде лежит яйцо $(1, 2)$, во втором — $(2, 4)$, в пятом — $(5, 1)$, а в третьем и четвёртом яиц нет.

Яйцо $(1, 2)$ добавить можно, несмотря на то что подобное яйцо на дереве уже есть, это приведёт к перекладыванию имеющегося яйца $(1, 2)$ в другое гнездо.

Также в начальную конфигурацию можно добавить любое из 10 яиц, существующих для дерева с пятью гнёздами, и каждое яйцо можно положить в любое из двух гнёзд, ему отвечающих, и для любого из добавляемых яиц и гнёзд это потребует конечное количество шагов. Таким образом, ответ на второй запрос — 20.

В результате следующего запроса яйцо $(1, 2)$ будет добавлено реально, и распределение яиц будет таким: в первом гнезде лежит яйцо $(1, 2)$, во втором — также $(1, 2)$, в четвёртом — $(2, 4)$, в пятом $(5, 1)$.

Теперь уже можно добавить только яйца $(1, 3)$, $(2, 3)$, $(4, 3)$ и $(5, 3)$, причём по-прежнему любое яйцо можно положить в каждое из двух упомянутых на нём гнёзд, поэтому ответ на запрос — 8.

Яйцо $(4, 2)$ добавить на дерево нельзя, поэтому состояние гнёзд не изменится.

Для добавления яйца $(5, 3)$ понадобится 5 перекладываний яиц, а после этого никакое новое яйцо за конечное количество шагов добавить уже нельзя.

Задача L. Рекурсивная схема

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В схеме рекурсивной микросхемы имеется N точек контактов, причем некоторые пары контактов соединены напрямую проводами. Кроме того, имеется всего S подсистем внутри схемы, каждая из которых является точной копией рассматриваемой схемы.

В схеме цепи есть три типа контактов:

1. Входные контакты цепи (K контактов). Это единственные контакты, соединяющие цепь со внешними проводящими путями.

2. Входные контакты вложенных подсхем ($S \cdot K$ контактов).

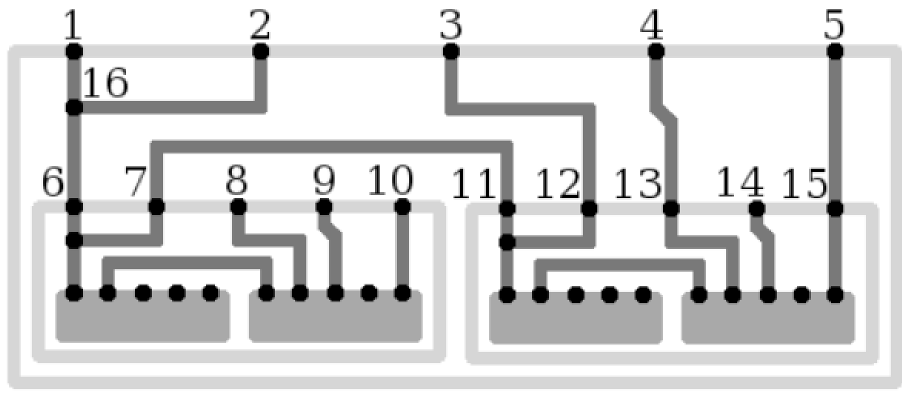
3. Вспомогательные контакты.

Все эти контакты могут быть связаны друг с другом проводами без каких-либо ограничений.

Сигналы распространяются по проводам. Когда сигнал достигает контакта, он может следовать по любому проводу, связанному с этим контактом. Если внешний сигнал достигает входной контакт подсхемы, он может войти в подсхему и двигаться дальше по ее проводам. Если внутренний сигнал достигает входной контакт подсхемы, он может выйти из подсхемы (если есть провод снаружи, и если внешняя цепь сама является подсхемой другой цепи).

Рассмотрим самую внешнюю схему. Определите, связаны ли два контакта путями. Контакты связаны путём, если сигнал может пройти по проводам от одного контакта от другого, возможно входя в ряд различных подсхем конечное количество раз.

Помимо факта подключения, в некоторых группах тестов от вас будет требоваться выяснить, насколько глубоко сигнал должен попасть в подсхемы, чтобы достичь одного контакта от другого. Внешняя цепь имеет глубину вложения 0; для её подсхем глубина вложения равна 1, а их подсхемы, в свою очередь, имеют глубину вложения 2 и т. д. Для произвольного пути сигнала, критической глубиной называется самая глубокая подсхема, через которую проходит путь. Определить минимальное значение критической глубины для пути между двумя заданными входными контактами внешней цепи.



Формат входных данных

Первая строка содержит пять целых чисел: N — количество контактов в схеме, K — количество входных контактов цепи, S — количество подсхем в цепи, M — число проводов в схеме цепи, T — номер группы тестов ($1 \leqslant K \leqslant 100\,000$, $0 \leqslant S \leqslant 1\,000$, $K \cdot (S+1) \leqslant N \leqslant 100\,000$, $0 \leqslant M \leqslant 100\,000$).

Следующие M строк определяют провода в схеме цепи. Каждый провод определяется двумя целыми числами a и b — номера контактов, напрямую связанных этим проводом ($1 \leqslant a \neq b \leqslant N$).

Контакты в схеме пронумерованы в порядке от 1 до N. Входные контакты пронумерованы от 1 до K. Входные контакты подсистемы t пронумерованы от $t \cdot K + 1$ до $t \cdot K + K$ (для $1 \leqslant t \leqslant S$).

j-й входной контакт на схеме t-й цепи является $(t \cdot K + j)$-ым контактом на схеме внешней схемы. Остальные контакты, если таковые существуют, являются вспомогательными.

Следующая строка содержит целое число Q — количество запросов ($1 \leqslant Q \leqslant 100\,000$). Каждый из остальные Q строк содержат один запрос, который нуждается в ответе. Каждый запрос определяется двумя целыми числами u и v — номера входных контактов внешней цепи ($1 \leqslant u \neq v \leqslant K$).

Формат выходных данных

В выходном файле выведите Q целых чисел, по одному числу в строке. i-е число должно быть ответом к i-му запросу: глубина вложения, необходимая для перехода от одного из входных контактов к другому. Если нет пути между двумя входными контактами, выведите число -1 вместо значения глубины.

В некоторых группах вам не надо выяснять, глубину сигнала. В этом случае для i-го запропса выведите -1 если от одного контакта нельзя добраться до другого и любое неотрицательное число, если путь между этими двумя контактами существует.

В тестах из условия требуется узнать глубину.

Система оценки

Ниже предоставлены критерии оценки:

№	Баллы	Ограничения	Особые случаи	Необх. группы
0	0	—	Тесты из условия	—
1	10	$N \leqslant 1000$, $M \leqslant 1000$, $S = 1$, $Q = 1$	—	—
2	10	$N \leqslant 1000$, $M \leqslant 1000$, $S = 1$	Не требуется узнавать глубину	—
3	15	$N \leqslant 1000$, $M \leqslant 1000$, $S = 1$	—	1, 2
4	15	$N \leqslant 1000$, $M \leqslant 1000$	Не требуется узнавать глубину	2
5	15	—	Не требуется узнавать глубину	2, 4
6	10	$N \leqslant 1000$, $M \leqslant 1000$, $Q = 1$	—	1
7	10	$N \leqslant 1000$, $M \leqslant 1000$	—	0 – 4, 6
8	15	—	—	0 – 7

Пример

стандартный ввод	стандартный вывод
16 5 2 7 0 1 16 2 16 6 16 7 11 3 12 4 13 5 15 4 1 2 2 3 3 4 4 5	0 1 2 -1

Задача M. Rooted MST

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Вам дан простой неориентированный граф содержащий $n+1$ вершин пронумерованных $0, 1, \ldots, n$ и $n + m$ ребер.

Вес ребра между вершинами 0 и i равен a_i для $1 \leqslant i \leqslant n$.

Вес ребра между вершинами u_i и v_i равен w_i для $1 \leqslant i \leqslant m$.

Вам нужно ответить на q запросов, каждый запрос содержит два целых числа i, w, вам нужно поменять вес ребра между вершинами 0 и i на w и найти вес минимального остовного дерева в графе.

Обратите внимание, что запросы перманенты, т.е. изменения остаются навсегда.

Формат входных данных

В первой строке записаны два целых числа n, m ($2 \leqslant n \leqslant 300\,000, 0 \leqslant m \leqslant 300\,000$).

Во второй строке записаны n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$).

В каждой из следующих m строк записаны три целых числа u_i, v_i, w_i ($1 \leqslant u_i, v_i \leqslant n, 0 \leqslant w_i \leqslant 10^9$).

Гарантируется, что данный граф простой, а именно не содержит петель и кратных ребер.

В следующей строке записано одно целое число q ($1 \leqslant q \leqslant 300\,000$).

В каждой из следующих q строк записаны два целых числа i, w ($1 \leqslant i \leqslant n, 1 \leqslant w \leqslant 10^9$).

Формат выходных данных

В каждой строке выведите одно целое число — вес минимального остовного дерева в графе после i запросов.

Система оценки

Подзадача	Баллы	Ограничения
1	10	$n, m, q \leqslant 2000$
2	10	Все веса равны 1 или 2
3	10	$w = 1$ во всех запросах
4	10	$i = 1$ во всех запросах
5	10	$i \leqslant 5$ во всех запросах
6	10	$m = n - 1, u_i = v_i - 1$
7	20	$n, m, q \leqslant 150\,000$
8	20	Нет дополнительных ограничений

Пример

стандартный ввод	стандартный вывод
5 7 3 2 1 2 1 1 5 1 1 3 2 2 5 2 4 5 2 3 4 1 2 4 2 1 2 1 10 3 2 2 3 4 1 3 2 5 1 5 3 3 1 2 3 4 3 5 1	6 6 5 5 5 6 6 6 6 5

Задача N. Блогеры-путешественники

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Ян и Татьяна решили стать блогерами-путешественниками и публиковать ролики о поездках по городам своей страны.

В стране есть n городов, пронумерованных от 1 до n. Город 1 — столица их страны. Города соединены m двусторонними дорогами, пронумерованными от 1 до m, каждая из которых соединяет два различных города. При этом одну и ту же пару городов могут соединять несколько различных дорог. Из любого города по дорогам можно доехать до любого другого города страны.

Путешественники планируют отправиться из столицы в какой-то другой город, но пока не выбрали в какой. Маршрут путешествия в город k будет состоять из городов $s_1, s_2, \ldots, s_q$ и дорог $r_1, r_2, \ldots, r_{q-1}$, таких что:

- $s_1 = 1$, $s_q = k$;
- дорога r_i соединяет города s_i и s_{i+1};
- ребята не проезжают по одной и той же дороге дважды, поэтому все r_i различны. Допускается проезжать несколько раз через один и тот же город, в том числе через город 1, где путешествие начинается, и город k, в котором путешествие заканчивается.

Для каждой дороги Ян и Татьяна посчитали длительность ролика, который получится при съемке путешествия по этой дороге, длительность ролика для дороги с номером i равна t_i.

В процессе путешествия каждый из ребят выберет одну из дорог маршрута и снимет ролик, посвящённый этой дороге. При этом Ян любит снимать короткие ролики, поэтому выберет на маршруте дорогу с наименьшим значением t_i, а Татьяна предпочитает длинные ролики, поэтому выберет дорогу с наибольшим значением t_i.

Суммарная длина двух роликов будет равна $\min\limits_{1 \leqslant i \leqslant q-1} t_{r_i} + \max\limits_{1 \leqslant i \leqslant q-1} t_{r_i}$.

Ребята планируют выложить ролики на известную платформу, где большей популярностью пользуются короткие ролики, поэтому они хотят минимизировать суммарную длину двух роликов. Чтобы выбрать конечный город и маршрут для путешествия, блогеры хотят для каждого конечного города k подсчитать минимальную по всем возможным маршрутам из города 1 в город k суммарную длину двух роликов.

Формат входных данных

В первой строке даны два целых числа n, m ($2 \leqslant n \leqslant 300\,000$, $1 \leqslant m \leqslant 300\,000$) — количество городов и дорог.

Следующие m строк содержат описания дорог. В i-й из этих строк находятся три целых числа u_i, v_i, t_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$, $0 \leqslant t_i \leqslant 10^9$) — номера городов, соединённых дорогой, и длительность ролика про эту дорогу.

Гарантируется, что по имеющимся дорогам можно проехать из любого города в любой другой, возможно, через другие города.

Формат выходных данных

Для каждого $2 \leqslant k \leqslant n$ выведите минимальную суммарную длину роликов Яна и Татьяны для путешествия, заканчивающегося в городе k.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 1 3 1 2 3 1	2 2
7 10 1 2 2 1 2 8 2 3 3 3 4 5 3 5 4 4 5 4 6 5 7 6 4 4 1 7 6 6 7 9	4 5 6 6 6 10
4 4 1 2 2 3 2 0 2 4 3 4 3 1	3 2 2

Замечание

В первом примере возможные оптимальные маршруты:

- $1 \overset{t=1}{\rightarrow} 3 \overset{t=1}{\rightarrow} 2$. Длина роликов в маршруте $1 + 1 = 2$.
- $1 \overset{t=1}{\rightarrow} 3$. Длина роликов в маршруте $1 + 1 = 2$.

Во втором примере возможные оптимальные маршруты:

- $1 \overset{t=2}{\rightarrow} 2$. Длина роликов в маршруте $2 + 2 = 4$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3$. Длина роликов в маршруте $2 + 3 = 5$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5 \overset{t=4}{\rightarrow} 4$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=3}{\rightarrow} 3 \overset{t=4}{\rightarrow} 5 \overset{t=4}{\rightarrow} 4 \overset{t=4}{\rightarrow} 6$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=8}{\rightarrow} 1 \overset{t=6}{\rightarrow} 7$. Длина роликов в маршруте $2 + 8 = 10$.

В третьем примере возможные оптимальные маршруты:

- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3 \overset{t=1}{\rightarrow} 4 \overset{t=3}{\rightarrow} 2$. Длина роликов в маршруте $0 + 3 = 3$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3$. Длина роликов в маршруте $0 + 2 = 2$.
- $1 \overset{t=2}{\rightarrow} 2 \overset{t=0}{\rightarrow} 3 \overset{t=1}{\rightarrow} 4$. Длина роликов в маршруте $0 + 2 = 2$.

Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи	Информация о проверке
		n	m	дополнительно		
1	9	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$m = n - 1$		первая ошибка
2	17	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$t_i = 0$ для всех дорог i из города 1		первая ошибка
3	12	$n \leqslant 300\,000$	$m \leqslant 300\,000$	$t_i = 10^9$ для всех дорог i из города 1		первая ошибка
4	9	$n \leqslant 10$	$m \leqslant 10$	каждая пара городов соединена не более чем одной дорогой		первая ошибка
5	6	$n \leqslant 20$	$m \leqslant 20$	каждая пара городов соединена не более чем одной дорогой	4	первая ошибка
6	6	$n \leqslant 2000$	$m \leqslant 2000$	$\lvert u_i - v_i \rvert = 1$ для всех дорог		первая ошибка
7	9	$n \leqslant 2000$	$m \leqslant 2000$		У, 4–6	первая ошибка
8	8	$n \leqslant 5000$	$m \leqslant 300\,000$		У, 4–7	только баллы
9	10	$n \leqslant 300\,000$	$m \leqslant 300\,000$	для всех a существует дорога между парой городов a и $a+1$; для любой пары дорог i и j, для которых $\lvert u_i - v_i \rvert = 1$ и $\lvert u_j - v_j \rvert > 1$ выполнено $t_i \leqslant t_j$	6	только баллы
10	14	$n \leqslant 300\,000$	$m \leqslant 300\,000$		У, 1–9	только баллы

Задача O. Кто убил Марка?

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мебибайт

Зачёркивал, стирал, неустанно толстел блокнот

Марк

У Марка есть изначально пустой граф на n вершинах.
У каждой вершины есть вес — неотрицательное целое число.
Также у Марка есть m троек целых чисел (a_i, b_i, s_i), где $1 \leqslant a_i \neq b_i \leqslant n$, и $s_i \geqslant 0$.
После чего Марк запускает следующий процесс:

- Если не существует такого i, что вершины a_i и b_i лежат в разных компонентах связности графа и суммарный вес вершин в компонентах связности a_i и b_i не меньше s_i, то процесс завершается.
- Если же такое i есть, то выбирается наименьшее такое i, и в граф добавляется ребро между вершинами a_i и b_i, и в блокнот записывается число i, после чего процесс повторяется уже на большем графе.

Однако в результате загадочных обстоятельств десятилетней давности Марк пропал вместе со своим блокнотом. Вам необходимо восстановить числа, записанные в блокноте.

Формат входных данных

В первой строке входных данных дано два целых положительных числа n и m $(1 \leqslant n, m \leqslant 3 \cdot 10^5)$ — количество вершин в графе и количество троек.

Вторая строка содержит n целых неотрицательных чисел w_i $(0 \leqslant w_i \leqslant 10^6)$ — веса вершин графа.

Последующие m строк описывают тройки. Каждая строка содержит по три целых неотрицательных числа a_i, b_i и c_i $(1 \leqslant a_i \neq b_i \leqslant n,\ 0 \leqslant s_i \leqslant 10^6)$ — описание очередной тройки.

Формат выходных данных

В первой строке выведите одно целое число k — количество чисел, записанных в блокноте после окончания процесса.

Во второй строке выведите k целых чисел, которые записаны в блокноте (в правильном порядке).

Примеры

стандартный ввод	*стандартный вывод*
5 5 1 4 3 4 0 4 5 5 3 1 1 2 5 2 4 3 1 4 1 4	4 2 3 1 4
3 5 3 2 2 1 2 6 1 2 6 1 2 3 1 2 6 2 3 6	2 3 5