

Задача А. 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 ($1 \leq n, m \leq 2000$) — количество строк и столбцов в лабиринте, соответственно.

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

Третья строка содержит два целых числа x, y ($0 \leq x, y \leq 10^9$) — 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

стандартный ввод	стандартный вывод
4 5 3 2 1 2***. ...** *.....	10
5 5 5 4 3 1 **... **.*. ...*. .***.	16

Задача В. 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 ($1 \leq N \leq 25\,000, 1 \leq A, B \leq 10^9$), 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	7671393960
0 0 0 0	

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 С. 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 положительных числа N , M и Q ($1 \leq N, Q \leq 100\,000$, $1 \leq M \leq N$) — количество городов, длительность строительства дорог и количество запросов соответственно.

В следующих Q строках вводятся по два целых числа A и B ($1 \leq A, B \leq N$) — номера городов двух математиков, которым интересно, через сколько дней они смогут встретиться (добраться из одного города 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 $N \leq 1000$, 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 четвертого дня, города, 20 и 9 связаны (через город 15).

Задача D. Подсчет опыта

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В очередной онлайн игре игроки, как обычно, сражаются с монстрами и набирают опыт. Для того, чтобы сражаться с монстрами, они объединяются в кланы. После победы над монстром, всем участникам клана, победившего его, добавляется одинаковое число единиц опыта. Особенностью этой игры является то, что кланы никогда не распадаются и из клана нельзя выйти. Единственная доступная операция — объединение двух кланов в один.

Поскольку игроков стало уже много, вам поручили написать систему учета текущего опыта игроков.

Формат входных данных

В первой строке входного файла содержатся числа n ($1 \leq n \leq 200000$) и m ($1 \leq m \leq 200000$) — число зарегистрированных игроков и число запросов.

В следующих m строках содержатся описания запросов. Запросы бывают трех типов:

- `join X Y` — объединить кланы, в которые входят игроки X и Y (если они уже в одном клане, то ничего не меняется).
- `add X V` — добавить V единиц опыта всем участникам клана, в который входит игрок X ($1 \leq V \leq 100$).
- `get X` — вывести текущий опыт игрока X .

Изначально у всех игроков 0 опыта и каждый из них состоит в клане, состоящим из него одного.

Формат выходных данных

Для каждого запроса `get X` выведите текущий опыт игрока X .

Пример

стандартный ввод	стандартный вывод
3 6	150
add 1 100	0
join 1 3	50
add 1 50	
get 1	
get 2	
get 3	

Задача Е. 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 ($1 \leq n \leq 150\,000$, $0 \leq m \leq 150\,000$, $1 \leq k \leq n$, $1 \leq s \leq 10^9$) — количество городов, количество дорог, количество заправок и объем бака.

В следующих m строках записаны по три целых числа u_i, v_i, c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq c_i \leq s$) — 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 ($1 \leq p_i \leq n$) — номера городов с заправками. Гарантируется, что заправка присутствует в городе 1.

Формат выходных данных

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

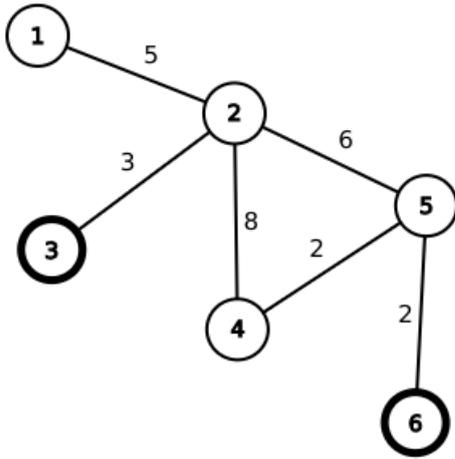
Во второй строке выведите x целых чисел — номера подходящих городов **в порядке возрастания**.

Пример

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

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	–	$n \leq 100$	–	0	
2	11	$k = 1$	$n \leq 5000$	–	–	
3	8	$k = 1$	–	–	2	
4	12	–	$n \leq 5000$	$c_i = 1$	–	
5	9	–	$n \leq 5000$	$c_i \leq 10$	0, 4	
6	17	–	$n \leq 5000$	–	0, 1, 2, 4, 5	
7	28	–	–	–	0–6	Offline-проверка.

Задача F. 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 пронумерованы 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 Иксевщины 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 ($1 \leq N_x, M_x \leq 5 \cdot 10^4$) — 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, соединенных 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 ($1 \leq N_y, M_y \leq 5 \cdot 10^4$) — 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 G. 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:

- (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.
- (Практический) Закончится ли процесс, если подложить яйцо (x, y) в гнездо x ? Если процесс закончится, то яйцо **добавляется** в реальности согласно описанному процессу.
- (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 , ($2 \leq n \leq 200\,000$, $0 \leq m \leq n$, $1 \leq q \leq 600\,000$), 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 проходятся в течение 10 минут. Пусть t_1 — количество запросов третьей группы, а t_2 и t_3 — количество запросов второй и первой групп соответственно. Если t_1, t_2, t_3 удовлетворяют ограничениям, то ответ на запрос считается найденным. Если же нет, то ответ считается не найденным. Если же количество запросов превышает заданное, то ответ считается не найденным.

Группа	Баллы	Дополнительные ограничения				Необх. группы	Комментарий
		n	t_1	t_2	t_3		
0	0	–	–	–	–	–	Приоритет
1	13	$n \leq 2000$	$t_1 \leq 2000$	$t_2 = 0$	$t_3 = 0$	–	
2	14	$n \leq 2000$	$t_1 \leq 2000$	$t_2 = 0$	$t_3 \leq 1$	1	
3	12	$n \leq 2000$	$t_1 \leq 2000$	$t_2 \leq 2000$	$t_3 \leq 2000$	0 – 2	
4	12	–	$t_1 \leq 2 \cdot 10^5$	$t_2 = 0$	$t_3 = 0$	1	
5	18	–	$t_1 \leq 2 \cdot 10^5$	$t_2 = 0$	$t_3 \leq 1$	1 – 2, 4	
6	31	–	$t_1 \leq 2 \cdot 10^5$	$t_2 \leq 2 \cdot 10^5$	$t_3 \leq 2 \cdot 10^5$	0 – 5	Offline-платформа

Пример

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

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 Н. 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, \dots, 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 ($1 \leq t \leq 300\,000$) — количество тестовых примеров.

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

Вторая строка каждого тестового примера содержит n целых чисел x_1, x_2, \dots, x_n ($-300\,000 \leq x_i \leq 300\,000$) — 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	4
5	-35
1 2 3 4 5	
3	
10 45 10	

Задача I. Авиареформа

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

Берляндия — большая страна с развитой системой авиасообщения. Всего в стране есть n городов, которые исторически обслуживаются авиакомпанией Берляфлот. Авиакомпания выполняет двухсторонние рейсы между m парами городов, i -й из них соединяет города с номерами a_i и b_i и имеет цену c_i на перелёт в каждую из сторон.

Известно, что с помощью рейсов Берляфлота можно добраться от любого города до любого другого (возможно, с пересадками), а стоимость любого маршрута из нескольких стыковочных рейсов Берляфлота равна стоимости самого дорогого из них. Более формально, стоимость маршрута из города t_1 в город t_k с $(k - 2)$ -мя пересадками в городах $t_2, t_3, t_4, \dots, t_{k-1}$ равна максимуму из стоимостей рейсов из города t_1 в t_2 , из t_2 в t_3 , из t_3 в t_4 и так далее до рейса из t_{k-1} в t_k . Разумеется, все эти рейсы должны выполняться авиакомпанией Берляфлот.

Недавно в Берляндии начала работать новая авиакомпания S8 Airlines. Эта авиакомпания совершает двусторонние рейсы между всеми парами городов, между которыми нет рейсов Берляфлота. Таким образом, между каждой парой городов есть рейс либо Берляфлота, либо S8 Airlines.

Стоимости рейсов авиакомпании S8 Airlines рассчитываются следующим образом: для каждой пары городов x и y , между которыми выполняется рейс S8 Airlines, стоимость этого рейса равняется минимальной стоимости маршрута между городами x и y у Берляфлота в соответствии с описанным ранее ценообразованием.

Известно, что с помощью рейсов S8 Airlines можно добраться от любого города до любого другого с возможными пересадками, и, аналогично Берляфлоту, стоимость маршрута между любыми двумя городами стыковочными рейсами S8 Airlines равна стоимости самого дорогого рейса в этом маршруте.

Из-за увеличившейся конкуренции с S8 Airlines Берляфлот решил провести авиареформу и изменить стоимости своих рейсов. А именно, для i -го своего рейса между городами a_i и b_i Берляфлот хочет сделать стоимость этого рейса равной минимальной стоимости маршрута между городами a_i и b_i у авиакомпании S8 Airlines. Помогите менеджерам Берляфлота рассчитать новые стоимости рейсов.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке вводятся два целых числа t и g ($1 \leq t \leq 10\,000$, $0 \leq g \leq 8$) — число наборов входных данных и номер группы тестов, под дополнительные ограничения которой подходит данный тест. Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных вводятся два целых числа n и m ($4 \leq n \leq 200\,000$, $n - 1 \leq m \leq 200\,000$, $m \leq \frac{(n-1)(n-2)}{2}$) — число городов в Берляндии и число рейсов у Берляфлота.

В следующих m строках описываются рейсы Берляфлота. В i -й строке даны три целых числа a_i, b_i и c_i ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^9$) — номера городов, которые соединены i -м рейсом Берляфлота, и стоимость i -го рейса Берляфлота.

Гарантируется, что никакой рейс не соединяет город сам с собой, а никакие 2 рейса не соединяют одну и ту же пару городов. Гарантируется, что рейсами Берляфлота можно добраться от любого города до любого другого и что рейсами S8 Airlines можно добраться от любого города до любого другого.

Обозначим за N сумму значений n по всем наборам входных данных, и за M — сумму значений m по всем наборам входных данных. Гарантируется, что $N, M \leq 200\,000$.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите m целых чисел, i -е из которых должно быть равно стоимости i -го рейса Берляфлота после авиареформы.

Пример

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

Замечание

В примере в первом наборе входных данных авиакомпания S8 Airlines будет выполнять рейсы между парами городов: (1, 3), (1, 4) и (2, 4).

Стоимость рейса между городами 1 и 3 будет равна 2, так как минимальная стоимость маршрута Берляфлота равна 2 — маршрут состоит из рейса между городами 1 и 2 стоимостью 1 и рейса между городами 2 и 3 стоимостью 2, максимум из стоимостей равен 2.

Стоимость рейса между городами 1 и 4 будет равна 3, так как минимальная стоимость маршрута Берляфлота составляет 3 — маршрут состоит из рейса между городами 1 и 2 стоимостью 1, рейса между городами 2 и 3 стоимостью 2 и рейса между городами 3 и 4 стоимостью 3, максимум из стоимостей равен 3.

Стоимость рейса между городами 2 и 4 будет равна 3, так как минимальная стоимость маршрута Берляфлота составляет 3 — маршрут состоит из рейса между городами 2 и 3 стоимостью 2 и рейса между городами 3 и 4 стоимостью 3, максимум из стоимостей равен 3.

После авиареформы стоимость рейса Берляфлота между городами 1 и 2 будет составлять 3, так как минимальная стоимость маршрута S8 Airlines между этими городами составляет 3 — маршрут состоит из рейса между городами 1 и 4 стоимостью 3 и рейса между городами 2 и 4 стоимостью 3, максимум равен 3.

Стоимость рейса Берляфлота между городами 2 и 3 будет составлять 3, так как минимальная стоимость маршрута S8 Airlines между этими городами составляет 3 — маршрут состоит из рейса между городами 2 и 4 стоимостью 3, рейса между городами 1 и 4 стоимостью 3 и рейса между 1 и 3 стоимостью 2, максимум равен 3.

Стоимость рейса Берляфлота между городами 3 и 4 будет составлять 3, так как минимальная стоимость маршрута S8 Airlines между этими городами составляет 3 — маршрут состоит из рейса между городами 1 и 3 стоимостью 2 и рейса между городами 1 и 4 стоимостью 3, максимум равен 3.

Во втором наборе входных данных у авиакомпании S8 Airlines будут следующие рейсы: между городами 1 и 4 стоимостью 1, между городами 2 и 3 стоимостью 1, между городами 2 и 5 стоимостью 2, между городами 3 и 4 стоимостью 1 и между городами 3 и 5 стоимостью 2.

Система оценки

Тесты к этой задаче состоят из 8 групп. Баллы за каждую группу ставятся только при прохож-

дении всех тестов группы и всех тестов необходимых групп. Обратите внимание, что 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
		n	N	c_i		
0	0	–	–	–	–	Тесты из условия.
1	11	$n \leq 10$	$N \leq 10\,000$	–	0	
2	10	$n \leq 100$	$N \leq 10\,000$	–	0, 1	
3	11	$n \leq 1000$	$N \leq 10\,000$	$c_i \leq 2$	–	
4	12	$n \leq 1000$	$N \leq 10\,000$	–	0, 1, 2	
5	12	–	–	–	–	Во всех наборах входных данных $m = n - 1$
6	17	–	–	$c_i \leq 2$	3	
7	10	–	–	$c_i \leq 10$	3, 6	
8	17	–	–	–	0 – 7	Offline-проверка.

Задача J. 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:

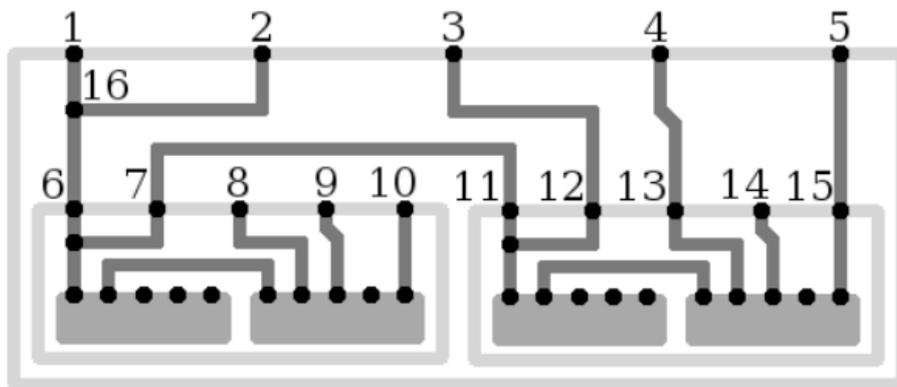
1. Входные контакты цепи (K контактов). Это единственные контакты, соединяющие цепь со внешними проводящими путями.
2. Входные контакты вложенных подсхем ($S \cdot K$ контактов).
3. Вспомогательные контакты.

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 к другому, 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 ($1 \leq K \leq 100\,000$, $0 \leq S \leq 1\,000$, $K \cdot (S + 1) \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$).

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

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 $1 \leq t \leq S$).

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 ($1 \leq Q \leq 100\,000$). Каждый из остальных Q строк содержат один запрос, который нуждается в ответе. Каждый запрос определяется двумя целыми числами u и v — номера входных контактов внешней цепи ($1 \leq u \neq v \leq K$).

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 запроса 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	$N \leq 1000, M \leq 1000, S = 1, Q = 1$	—	—
2	10	$N \leq 1000, M \leq 1000, S = 1$	Не требуется узнавать глубину	—
3	15	$N \leq 1000, M \leq 1000, S = 1$	—	1, 2
4	15	$N \leq 1000, M \leq 1000$	Не требуется узнавать глубину	2
5	15	—	Не требуется узнавать глубину	2, 4
6	10	$N \leq 1000, M \leq 1000, Q = 1$	—	1
7	10	$N \leq 1000, M \leq 1000$	—	0 – 4, 6
8	15	—	—	0 – 7

Пример

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