Задача A. Остовное дерево

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

Найти в связном графе остовное дерево минимального веса.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leqslant n \leqslant 20000, 0 \leqslant m \leqslant 100000$). Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \leqslant b_i, e_i \leqslant n, 0 \leqslant w_i \leqslant 100000$).

Граф является связным.

Формат выходных данных

Выведите единственное целое число — вес минимального остовного дерева.

Пример

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

Задача B. День Объединения

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

В Байтландии есть целых n городов, но нет ни одной дороги. Король страны, Вальдемар де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было добраться от любого города до любого другого. Когда строительство будет завершено, король планирует отпраздновать День Объединения. К сожалению, казна Байтландии почти пуста, поэтому король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 5\,000$) — количество городов в Байтландии. Каждая из следующих n строк содержит по два целых числа x_i, y_i — координаты i-го города ($-10\,000 \leqslant x_i, y_i \leqslant 10\,000$). Никакие два города не расположены в одной точке.

Формат выходных данных

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее 10^{-4}.

Пример

стандартный ввод	стандартный вывод
6 1 1 7 1 2 2 6 2 1 3 7 3	9.6568542495

Задача C. Разрезание графа

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

Формат входных данных

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n, количество рёбер m и количество операций k ($1 \leqslant n \leqslant 50\,000$, $0 \leqslant m \leqslant 100\,000$, $m \leqslant k \leqslant 150\,000$).

Следующие m строк задают рёбра графа; i-я из этих строк содержит два числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$), разделённые пробелами — номера концов i-го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа `cut` задаётся строкой «`cut` u v» ($1 \leqslant u, v \leqslant n$), которая означает, что из графа удаляют ребро между вершинами u и v. Операция типа `ask` задаётся строкой «`ask` u v» ($1 \leqslant u, v \leqslant n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

Формат выходных данных

Для каждой операции `ask` во входном файле выведите на отдельной строке слово «`YES`», если две указанные вершины лежат в одной компоненте связности, и «`NO`» в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

Пример

стандартный ввод	стандартный вывод
3 3 7 1 2 2 3 3 1 ask 3 3 cut 1 2 ask 1 2 cut 1 3 ask 2 1 cut 2 3 ask 3 1	YES YES NO NO

Задача D. Расстояние X

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

Вам дан неориентированный взвешенный граф из N вершин и M ребер. Назовем ценой пути между двумя вершинами вес максимального ребра на этом пути. Найдите количество пар с минимальной ценой пути между ними, равной X.

Формат входных данных

Первая строка содержит три целых числа N, M и X ($1 \leqslant N \leqslant 10^5, 1 \leqslant M \leqslant 3 \cdot 10^5, 1 \leqslant X \leqslant 10^9$). Следующие M строк содержат по три числа a_i, b_i и w_i, обозначающие ребро между вершинами a_i и b_i веса w_i. ($1 \leqslant w_i \leqslant 10^9$)

Формат выходных данных

Выведите одно число — ответ на задачу

Примеры

стандартный ввод	стандартный вывод
7 6 3 1 2 1 1 3 2 3 4 3 4 5 1 4 6 2 1 7 4	9
8 8 4 1 3 2 2 4 1 1 5 1 6 7 3 5 8 4 8 4 4 6 5 5 7 8 6	11

Задача E. Шарады

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

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 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 ($1 \leqslant N, Q \leqslant 100000, 1 \leqslant M \leqslant N$) — 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.

Пример

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

Задача F. Всем чмоки в этом чатике!

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

Сегодня Мэри, как программисту социальной сети «Телеграфчик», предстоит реализовать сложную систему управления чатами.

Задача Мэри усложняется тем, что в социальную сеть «Телеграфчик» внедрена продвинутая система шифрования «ZergRus», простая, как всё гениальное. Суть её в том, что в системе хранится одна переменная zerg, которая принимает значения от 0 (включительно) до $p = 10^6 + 3$ (исключая p) и меняется в зависимости от событий в системе.

В социальной сети всего n пользователей ($1 \leqslant n \leqslant 10^5$). В начале дня каждый пользователь оказывается в своём собственном чате, в котором больше никого нет. Переменная zerg в начале дня устанавливается равной 0.

В течение дня происходят события типов:

1. Участник с номером $(i + zerg) \bmod n$ посылает сообщение всем участникам, сидящим с ним в чате (в том числе и себе самому), при этом переменная zerg заменяется на $(30 \cdot zerg + 239) \bmod p$.
2. Происходит слияние чатов, в которых сидят участники с номерами $(i + zerg) \bmod n$ и $(j + zerg) \bmod n$. Если участники и так сидели в одном чате, то ничего не происходит. Если в разных, то чаты объединяются, а переменной zerg присваивается значение $(13 \cdot zerg + 11) \bmod p$.
3. Участник с номером $(i + zerg) \bmod n$ хочет узнать, сколько сообщений он не прочитал, и прочитать их. Если участник прочитал q новых сообщений, то переменной zerg присваивается значение $(100\,500 \cdot zerg + q) \bmod p$.

Вы поможете Мэри реализовать систему, обрабатывающую эти события?

Формат входных данных

В первой строке входного файла записаны натуральные числа n ($1 \leqslant n \leqslant 10^5$) — число пользователей социальной сети. и m ($1 \leqslant m \leqslant 3 \cdot 10^5$) — число событий, произошедших за день. В следующих m строках содержится описание событий. Первое целое число в строке означает тип события t ($1 \leqslant t \leqslant 3$). Если $t = 1$, далее следует число i ($0 \leqslant i < n$), по которому можно вычислить, какой участник послал сообщение. Если $t = 2$, далее следуют числа i и j ($0 \leqslant i, j < n$), по которым можно вычислить номера участников, чаты с которыми должны объединиться. Если $t = 3$, далее следует число i ($0 \leqslant i < n$), по которому можно вычислить номер участника, желающего узнать, сколько у него сообщений, и прочитать их.

Формат выходных данных

Для каждого события типа 3 нужно вывести число непрочитанных сообщений у участника.

Пример

стандартный ввод	стандартный вывод
4 10 1 0 1 2 1 1 1 2 3 1 2 1 2 1 3 3 3 2 3 2 3 2	1 1 2

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, $(2 \leqslant n \leqslant 200000, 0 \leqslant m \leqslant n, 1 \leqslant q \leqslant 600000)$, 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.

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

Задача H. Налоги

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

Король Байтландии следует мировой тенденции и вводит налоги везде, где только может. Недавно он придумал налог на путешествия, который взымается с каждого, кто путешествует по стране.

Каждой Байтландской дороге сопоставлен размер налога за нее. При прохождении через город путешественнику необходимо заплатить налог, равный максимуму из размеров налога за дорогу, по которой он вошел в город, и дорогу, по которой он вышел из города. Также необходимо заплатить за первый и последний город: для них налог равен единственной соответствующей дороге.

Ваш друг Байтазар собирается в путешествие из Байтгорода в Байтополис. Помогите ему спланировать маршрут таким образом, чтобы размер уплаченного налога был минимален.

Формат входных данных

Первая строка ввода содержит два целых числа n и m ($2 \leqslant n \leqslant 100\,000, 1 \leqslant m \leqslant 200\,000$) — количество городов и количество дорог в Байтландии. Города пронумерованы от 1 до n.

Следующие m строк содержат описания дорог. i-ая из этих строк содержит три целых числа a_i, b_i, c_i ($1 \leqslant a_i, b_i \leqslant n, a_i \neq b_i, 1 \leqslant c_i \leqslant 1\,000\,000$). Это означает, что города a_i и b_i соединены двунаправленной дорогой с размером налога, равным c_i байтлей. Каждая пара соединена не более, чем одной дорогой.

Формат выходных данных

В первой и единственной строке вывода должно содержаться одно целое число — минимальный размер налога (в байтлях) на путешествие из Байтгорода (т.е. города номер 1) в Байтополис (т.е. город номер n). Гарантируется, что существует путь, соединяющий эти два города.

Примеры

стандартный ввод	стандартный вывод
4 5 1 2 5 1 3 2 2 3 1 2 4 4 3 4 8	12
6 6 6 2 4 4 2 3 3 5 3 2 1 8 4 1 2 4 6 6	13

Задача I. Школы

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

С целью подготовки к проведению олимпиады по информатике мэр решил обеспечить надежным электроснабжением все школы города. Для этого необходимо провести линию электропередач от альтернативного источника электроэнергии "Майбутя"к одной из школ города (к какой неважно), а также соединить линиями электропередач некоторые школы между собой.

Считается, что школа имеет надежное электроснабжение, если она напрямую связана с источником "Майбутя либо с одной из тех школ, которые имеют надежное электроснабжение.

Известна стоимость соединения между некоторыми парами школ. Мэр города решил выбрать одну из двух наиболее экономичных схем электроснабжения (стоимость схемы равняется сумме стоимостей соединений пар школ).

Напишите программу, которая вычисляет стоимость двух наиболее экономных схем альтернативного электроснабжения школ.

Формат входных данных

В первой строке входного файла находятся два натуральных числа, разделенных пробелом: N ($3 \leqslant N \leqslant 100$), количество школ в городе, и M — количество возможных соединений между ними.

В каждой из последующих M строк находятся по три числа: A_i, B_i, C_i, разделенных пробелами, где C_i — стоимость прокладки линии электроснабжения ($1 \leqslant Ci \leqslant 300$) от школы A_i до школы B_i.

Формат выходных данных

В единственной строке выходного файла должны содержаться два натуральных числа S_1 и S_2, разделенных пробелом — две наименьшие стоимости схем ($S_1 \leqslant S_2$). $S_1 = S_2$ тогда и только тогда, когда существует несколько схем надежного электроснабжения наименьшей стоимости.

Гарантируется, что для входных данных существует две различные схемы надёжного электроснабжения.

Пример

стандартный ввод	стандартный вывод
5 8 1 3 75 3 4 51 2 4 19 3 2 95 2 5 42 5 4 31 1 2 9 3 5 66	110 121

Задача J. Постройка дорожной сети

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 мегабайт

Странствующий торговец прибыл в королевство Lazyland. Он планирует начать свое путешествие из столицы, посетить каждый из городов королевства, а затем вернуться в столицу. Торговец знает координаты каждого города, однако дороги еще не были построены. Очевидно, не имея дорог, не возможно посетить все города королевства!

Известно, что в королевстве есть n городов. Их координаты равны: (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n)$. Столица находится в точке (x_1, y_1). Король выразил свое желание помочь торговцу. Он прикажет построить для него дороги. Однако, жители страны слишком ленивые.

Как и все жители страны, инженеры тоже очень ленивы, поэтому если потребуется построить дорогу от точки (x_i, y_i) до точки (x_j, y_j), они будут использовать только горизонтальные или вертикальные отрезки. Поэтому нетрудно заметить, что длина такой дороги получится равной $|x_i - x_j| + |y_i - y_j|$.

Так как рабочие тоже невероятно ленивы, они согласятся построить не более, чем $n - 1$ дорогу для торговца — они слышали, что $n - 1$ дороги достаточно, чтобы путешествовать между всеми городами. Концами каждой дороги по понятным причинам будут какие-либо два различных города.

Торговцу дали право выбрать, какие именно дороги должны быть построены. Помогите ему вычислить длину минимального маршрута, который начинается в столице, обходит все города и затем возвращается в столицу при условии, что можно построить любые $n - 1$ дороги.

Формат входных данных

В первой строке записано число n ($1 \leqslant n \leqslant 10^4$) — количество городов в королевстве.

В каждой из следующих n строк записаны два числа x_i и y_i ($-1000 \leqslant x_i, y_i \leqslant 1000$) — координаты городов. Координаты столицы записаны в первой из этих строк.

Гарантируется, что никакие два города не находятся в одной точке.

Формат выходных данных

Выведите длину кратчайшего маршрута торговца, удовлетворяющего описанным в задаче требованиям.

Примеры

стандартный ввод	стандартный вывод
3 1 1 2 2 3 3	8
4 2 1 -1 2 -2 -1 1 -2	24
6 1 2 2 3 2 2 3 4 4 3 3 1	16

Задача K. Пифагоров экспресс (для C++)

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

В стране Флатландии есть n городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами x_1, y_1 до города с координатами x_2, y_2 за время $(x_1 - x_2)^2 + (y_1 - y_2)^2$. По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.

Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером s. Вам же хочется попасть в город с номером t. Определите, за какое наименьшее время это можно сделать.

Формат входных данных

В первой строке задано число городов n ($1 \leqslant n \leqslant 2 \cdot 10^4$). В следующих n строках заданы координаты городов x_i, y_i ($|x_i|, |y_i| \leqslant 10^4$). В последней строке даны s и t — номера начального и конечного города в пути ($1 \leqslant s, t \leqslant n$).

Формат выходных данных

Выведите одно число — минимальное время, за которое можно добраться из s в t.

Пример

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

Задача L. Телепорты

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

Маленький Эдсгер живёт на прямой в точке S, и хочет попасть к своему другу Роберту в точку T. На прямой есть n телепортов, i-й телепорт находится в точке x_i и может мгновенно перенести на расстояние ровно d_i (то есть в точки $x_i - d_i$ и $x_i + d_i$). Кроме использования телепортов, можно ходить по прямой пешком со скоростью 1. За какое минимальное время можно дойти из S в T?

Обратите внимание, что Эдсгер не обязан использовать телепорт, если находится в одной с ним точке.

Формат входных данных

В первой строке записаны три целых числа n, S и T ($0 \leqslant n \leqslant 10^5$, $0 \leqslant S, T \leqslant 10^9$).

В следующих n строках описаны телепорты по одному в строке. Описание каждого телепорта состоит из двух целых чисел x_i, d_i ($0 \leqslant x_i \leqslant 10^9, 1 \leqslant d_i \leqslant 10^9$).

Гарантируется, что все телепорты находятся в разных точках.

Формат выходных данных

Выведите одно число — минимальное время, необходимое для того, чтобы добраться из S в T.

Примеры

стандартный ввод	стандартный вывод
0 10 20	10
1 10 20 11 8	2
1 20 10 11 8	10
1 10 20 9 9	3
1 10 20 15 1000	10
2 10 20 10 1000 1009 988	2

Задача M. Скандинавские контрабандисты

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

После того, как ограничения на приграничную торговлю между Норвегией и Швецией были сняты, многие предприниматели стали пользоваться возможностью торговать своими товарами с соседней страной. Однако, власти обеспокоены тем, что отсутствие ограничений поспособствует росту контрабанды товаров на границе. Для предотвращения данной угрозы таможенные службы Норвегии и Швеции объединились в борьбе с общим врагом.

Для того, чтобы успешно пройти таможню, необходимо посетить некоторое количество КПП. Всего на границе есть n КПП, пронумерованных числами от 1 до n, где КПП под номером 1 является входом на таможню, а КПП с номером n — выходом уже по другую сторону границы. Также были построены m дорог с двусторонним движением, которые соединяют некоторые пары КПП. Для прохождения i-го КПП требуется t_i минут.

Каждый КПП может контролироваться либо норвежской, либо шведской таможенной службой, причем Норвегия должна контролировать k КПП, а Швеция, соответственно, $n-k$ КПП. Известно, что если некоторая дорога соединяет два КПП, контролируемых таможенной службой одной и той же страны, то все контрабандисты, перемещающиеся по данной дороге, обязательно будут пойманы, чего нельзя сказать об остальных дорогах. Также известно, что контрабандисты всегда спешат и хотят пройти таможню как можно быстрее, поэтому они всегда выбирают кратчайший по затратам времени путь из КПП с номером 1 в КПП с номером n.

От вас требуется распределить все КПП между таможенными службами Норвегии и Швеции таким образом, чтобы любой контрабандист, направляющийся из КПП с номером 1 в КПП с номером n, был пойман.

Формат входных данных

В первой строке записаны три целых числа n, m и k ($2 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 2 \cdot 10^5$, $0 \leqslant k \leqslant n$) — количество КПП, количество дорог, а также количество КПП, которые нужно отдать под управление норвежской таможенной службы, соответственно.

Во второй строке через пробел записаны n положительных чисел $t_1, t_2, \ldots, t_n$ ($1 \leqslant t_i \leqslant 10^4$) — количество времени, необходимое для прохождения каждого КПП.

В каждой из следующих m строк записаны два целых числа u и v ($1 \leqslant u, v \leqslant n$) — это означает, что между КПП с номерами u и v имеется дорога с двусторонним движением.

Гарантируется, что от любого КПП можно добраться до любого другого КПП, пользуясь только имеющимися дорогами. Между каждой парой КПП может быть не более одной дороги. Никакая дорога не соединяет КПП с самим собой.

Формат выходных данных

Если существует способ распределить все КПП между таможенными службами Норвегии и Швеции требуемым образом, выведите строку, состоящую из n символов «`N`» и «`S`», где символ s_i равен «`N`», если i-й КПП следует отдать Норвегии, и «`S`» — в противном случае.

Если требуемого распределения КПП не существует, выведите строку «`impossible`» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
3 2 0 1 1 1 1 2 2 3	SSS
2 1 1 1 1 1 2	impossible
8 9 4 3 3 1 2 2 3 2 1 1 2 1 3 1 4 2 5 3 6 4 7 5 8 6 8 7 8	NNNNSSSS

Задача N. Очередная задача про графы

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

Сергей играет в одну из своих любимых настольных игр. Если формализовать правила игры, то получится следующая задача.

Дан ориентированный граф, состоящий из N вершин и M ребер. Ребро с номером i соединяет две различные вершины U_i и V_i и имеет длину W_i. При помощи i-го ребра можно переместиться из вершины U_i в вершину V_i, но не наоборот.

Изначально Сергей должен поместить обе свои руки в две различные вершины. За один ход он может переместить одну руку в некоторую вершину, соседнюю с той, где изначально находилась рука, воспользовавшись некоторым ребром. Для перемещения руки из вершины U_i в вершину V_i Сергей тратит W_i секунд. Обратите внимание, что одновременно Сергей может перемещать только одну руку. Игра заканчивается, когда обе руки Сергея оказываются в одной и той же вершине.

Сергей хочет узнать некоторую информацию об этой игре. Для каждого p ($2 \leqslant p \leqslant N$) ему необходимо вычислить минимальное количество секунд, необходимых, чтобы закончить игру при условии, что изначально его руки находятся в вершинах 1 и p. Обратите внимание, что для некоторых p закончить игру может быть невозможно.

Формат входных данных

Первая строка содержит два целых числа N и M ($2 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 2 \cdot 10^5$) — количество вершин и ребер в графе.

Каждая из следующих M строк содержит три целых числа U_i, V_i и W_i ($1 \leqslant U_i, V_i \leqslant N$, $U_i \neq V_i$, $1 \leqslant W_i \leqslant 10^9$) — описание очередного ориентированного ребра, соединяющего вершины U_i и V_i, имеющего длину W_i. Гарантируется, что в графе нет пары ребер i и j ($i \neq j$), таких что $U_i = U_j$ и $V_i = V_j$.

Формат выходных данных

Выведите $N - 1$ целых чисел: i-е число должно равняться минимальному количеству секунд, за которые можно закончить игру при условии, что изначально руки Сергея находятся в вершинах 1 и $i + 1$. Если для какого-то i закончить игру не удастся, выведите число -1.

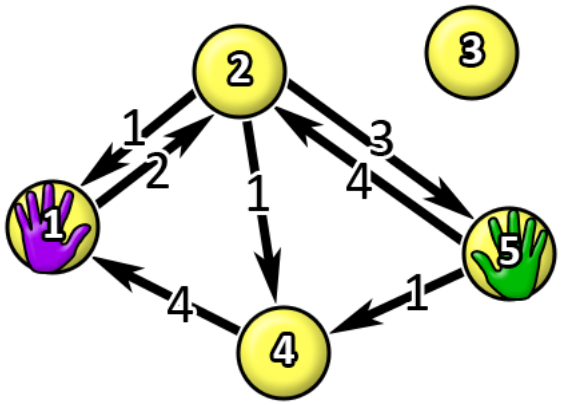
Пример

стандартный ввод	стандартный вывод
5 7 1 2 2 2 4 1 4 1 4 2 5 3 5 4 1 5 2 4 2 1 1	1 -1 3 4

Замечание

Если изначально руки Сергея находятся в вершинах 1 и 5, можно выполнить следующие действия:

1. Переместить вторую руку в вершину 4 за 1 секунду.
2. Переместить первую руку в вершину 2 за 2 секунды.
3. Переместить первую руку в вершину 4 за 1 секунду.



Суммарное затраченное время равно $1 + 2 + 1 = 4$ секунды.

Задача O. Пляж

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

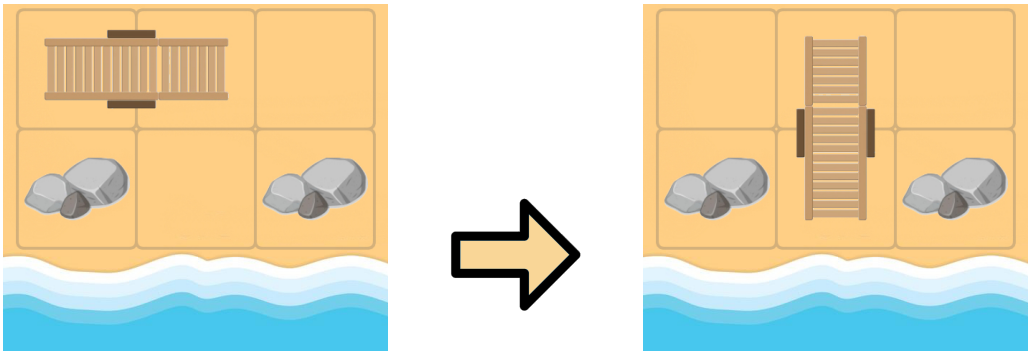
Андрей обожает море. Именно поэтому в разгар летнего сезона он решил отправиться на пляж, взяв с собой лежак, чтобы позагорать.

Пляж представляет собой прямоугольное поле, состоящее из n строк и m столбцов. Некоторые клетки этого поля свободны, на некоторых расположены дорожки, камни, ларьки и другие несдвигаемые объекты, а так же на двух соседних по стороне клетках могут стоять лежаки, расположенные как горизонтально, так и вертикально.

Андрей надеется поставить где-то свой лежак, но вот незадача, свободных мест для него может уже не быть! Именно поэтому Андрей просит вас помочь найти ему свободное место для лежака. Лежак Андрея тоже должен занимать две соседние по стороне клетки.

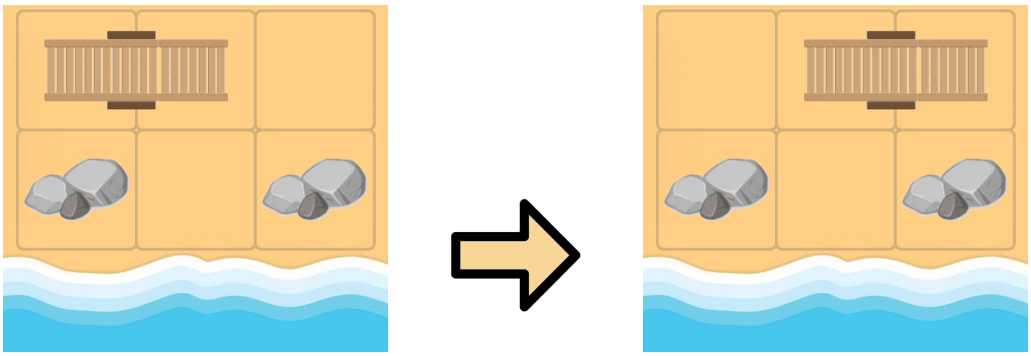
Если двух соседних свободных клеток нет, то для того, чтобы освободить место для лежака, придется потревожить других туристов. Для этого вы можете делать следующие действия:

- Подойти к любому лежаку, и доставив p единиц дискомфорта его хозяину, поднять лежак за одну из сторон и повернуть на 90 градусов. Одна из половин лежака должна остаться в той же клетке, а вторая переместиться в свободную клетку. При этом во время поворота на пути лежака может быть что угодно.



Поворот лежака на 90 градусов вокруг клетки $(1, 2)$.

- Подойти к любому лежаку, и доставив q единиц дискомфорта его хозяину, подвинуть лежак вдоль его длинной стороны на одну клетку. Одна сторона лежака должна переместиться на место другой, а другая — в свободную клетку.



Сдвиг лежака на одну клетку вправо.

В любой момент времени каждый лежак занимает две соседние клетки, не содержащие ничего другого. Вы можете двигать одновременно максимум один лежак.

Помогите Андрею освободить место для еще одного лежака, доставив минимальное количество дискомфорта окружающим, или скажите, что сделать это не получится.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leqslant n, m \leqslant 300\,000$, $1 \leqslant n \cdot m \leqslant 300\,000$) — количество строк и столбцов на поле.

Вторая строка содержит два целых числа p и q ($1 \leqslant p, q \leqslant 10^9$) — количество единиц дискомфорта, которые приносят поворот и сдвиг лежака, соответственно.

Каждая из следующих n строк содержит m символов, описывающие клетки поля. Все строки состоят из символов «L», «R», «D», «U», «.» и «#», задающих тип клетки. Символы «L», «R», «D» и «U» обозначают половину лежака, находящегося в этой клетке — левая, правая, нижняя или верхняя половина, соответственно. Символ «.» обозначает свободную клетку, а символ «#» — клетку, занятую несдвигаемым объектом.

Формат выходных данных

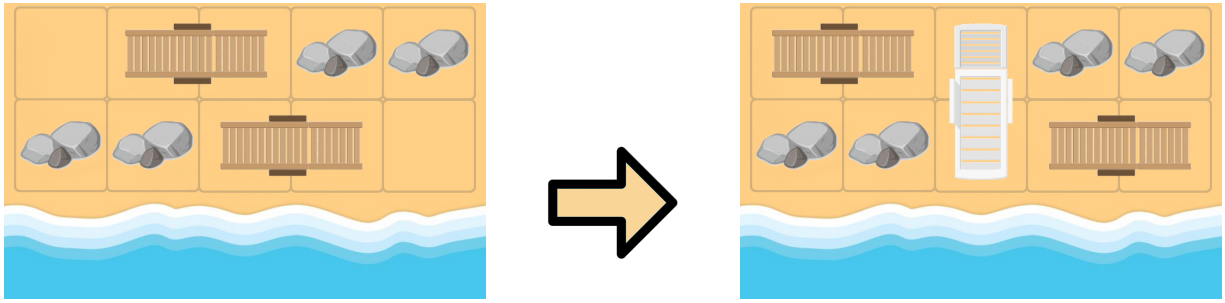
Выведите одно число — минимальное количество дискомфорта, которое возникнет при освобождении места для лежака. Если освободить место для лежака невозможно, выведите число -1.

Примеры

стандартный ввод	стандартный вывод
2 5 5 2 .LR## ##LR.	4
2 3 4 5 LR. #.#	-1
4 3 10 10 .LR ### UU# DD.	-1
3 6 10 7 .U##.# #DLR## .##LR.	24

Замечание

В первом примере, передвинув верхний лежак налево, а нижний лежак направо, Андрей сможет вертикально поставить лежак посередине пляжа. Таким образом, мы доставим $2 + 2 = 4$ единицы дискомфорта. Можно показать, что доставить меньшее количество дискомфорта не получится.



Оптимальные действия в первом примере (лежак Андрея обозначен белым цветом).

Во втором примере из условия невозможно освободить место для лежака Андрея. Всевозможные состояния пляжа после поворотов и сдвигов показаны в условии задачи.

Задача P. Доставка пиццы

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

Алиса живет в New Tsukuba City и учится в колледже. Завтра в городе начнется новый социальный эксперимент, связанный со сменой направлений движения на односторонних дорогах города. Каждый день рано утром направление одной их дорог будет сменено на противоположное, после чего поздно вечером того же дня направление будет изменено на начальное.

Более формально: всего в городе имеется m дорог, пронумерованных числами от 1 до m. В i-й день утром направление i-й дороги будет изменено на противоположное, а вечером направление i-й дороги станет исходным.

Каждый день Алиса заказывает пиццу из одной и той же пиццерии. Разумеется, пицца доставляется по кратчайшему маршруту от пиццерии до дома Алисы.

За счет смены направлений дорог кратчайший путь может меняться изо дня в день. Помогите Алисе понять, как проведение эксперимента повлияет на время доставки пиццы.

Формат входных данных

В первой строке записаны два числа n и m ($2 \leqslant n \leqslant 10^5, 1 \leqslant m \leqslant 10^5$) — количество перекрестков в городе и количество односторонних дорог соответственно.

Каждая из следующих m строк содержит три числа a_i, b_i и c_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$, $1 \leqslant c_i \leqslant 10^5$). Эти числа означают, что i-я дорога соединяет перекрестки с номерами a_i и b_i, причем можно проехать по ней в направлении $a_i \to b_i$ (и только в этом направлении) за время c_i.

Обратите внимание, что несколько различных дорог могут соединять два перекрестка.

Пиццерия находится на перекрестке с номером 1, а дом Алисы — на перекрестке с номером 2.

Гарантируется, что до начала эксперимента от пиццерии до дома Алисы существует маршрут.

Формат выходных данных

Выведите m строк. В i-й строке выведите:

- `HAPPY`, если кратчайший путь в i-й день станет короче;
- `SOSO`, если длина кратчайшего пути в i-й день не изменится;
- `SAD`, если в i-й день кратчайший путь станет длиннее, либо в i-й день не будет пути от пиццерии до дома Алисы.

Алисе не важно, сможет ли доставщик пиццы добраться от ее дома до пиццерии на обратном пути.

Примеры

стандартный ввод	стандартный вывод
4 5 1 3 5 3 4 6 4 2 7 2 1 18 2 3 12	SAD SAD SAD SOSO HAPPY
7 5 1 3 2 1 6 3 4 2 4 6 2 5 7 5 6	SOSO SAD SOSO SAD SOSO
10 14 1 7 9 1 8 3 2 8 4 2 6 11 3 7 8 3 4 4 3 2 1 3 2 7 4 8 4 5 6 11 5 8 12 6 10 6 7 10 8 8 3 6	SOSO SAD HAPPY SOSO SOSO SOSO SAD SOSO SOSO SOSO SOSO SOSO SOSO SAD

Задача Q. Липецкие дороги

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

Вася приехал на Вселипецкую олимпиаду школьников по информатике в город L. В городе есть n перекрестков, пронумерованных числами от 1 до n, некоторые из которых соединены двунаправленными дорогами. В городе L есть почти все: воздушное метро, аэропорт и, конечно, такси.

Всего в городе есть две фирмы такси. Чтобы заказать такси, необходимо открыть приложение фирмы и выбрать две точки — номера перекрестков, откуда и куда нужно добраться. Для каждой фирмы известно, по каким дорогам может перемещаться такси этой фирмы, а также за какое время оно может проехать каждую дорогу. Во всех такси есть навигатор, который прокладывает самый быстрый маршрут в точку назначения по доступным дорогам.

Все в этом городе хорошо, но интернет не везде работает хорошо. Поэтому заказать себе такси можно только на определенных перекрестках. Время на загрузку приложений различное для всех перекрестков. В самом отеле есть интернет, и там можно мгновенно выбрать приложение и заказать такси.

Вася проспал, и теперь опаздывает на первый тур. Помогите ему как можно быстрее добраться на такси от отеля до места проведения олимпиады. Так как Вася хочет добраться до места проведения олимпиады максимально быстро, он хочет действовать оптимально, поэтому иногда он может выходить из такси на некотором перекрестке, после чего вызывать новое такси и продолжать поездку на нем.

Отель и место проведения олимпиады находятся на разных перекрестках.

Формат входных данных

В первой строке записаны четыре целых числа: n, m_1, m_2 и k ($2 \leqslant n \leqslant 100\,000, 0 \leqslant m_1 + m_2 \leqslant 200\,000, 0 \leqslant k \leqslant n$) — количество перекрестков, дорог для первой фирмы, дорог для второй фирмы и перекрестков с работающим интернетом соответственно.

Далее в m_1 строках перечислены дороги, по которым может перемещаться такси первой фирмы по одной в строке. Каждая из дорог задается тремя числами a_i, b_i, w_i ($1 \leqslant a_i, b_i \leqslant n, 0 \leqslant w_i \leqslant 10^6$), где a_i, b_i — номера перекрестков, между которыми есть дорога, а w_i — время, за которое такси первой фирмы может преодолеть эту дорогу.

Далее в m_2 строках перечислены дороги для второй фирмы в аналогичном формате.

Далее в k строках перечислены номера домов ind_i, около которых работает интернет, и время загрузки приложений ($0 \leqslant c_i \leqslant 10^6$).

В последней строке находятся два числа s, f ($s \neq f$) — адреса отеля и места проведения олимпиады.

Между любой парой перекрестков существует не более одной дороги для каждого такси. Дорога не может соединять перекресток с самим собой.

Не гарантируется, что из любого перекрестка можно добрать до любого другого.

Формат выходных данных

Выведите единственное целое число — наименьшее возможное время прибытия к месту проведения олимпиады. Если же до места нельзя добраться, выведите `-1`.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	–
1	10	$2 \leqslant n \leqslant 5$	подзадача	–
2	10	$2 \leqslant n \leqslant 300$	подзадача	1
3	10	$2 \leqslant n \leqslant 1000, c_i = 0, k = n$	подзадача	–
4	10	$c_i = 0, k = n$	подзадача	3
5	20	$c_i = 1, w_i = 1$	подзадача	–
6	20	$2 \leqslant n \leqslant 1000, 0 \leqslant m_1 + m_2 \leqslant 20\,000$	подзадача	1, 2
7	20	Дополнительных ограничений нет	подзадача	1, 2, 3, 4, 5, 6

Примеры

стандартный ввод	стандартный вывод
7 6 6 2 1 2 10 1 3 1 2 3 1 2 4 3 3 5 5 4 6 3 1 5 10 2 3 2 3 4 5 4 5 10 4 7 30 6 7 100 2 6 5 1 1 7	45
3 1 1 0 1 2 1 2 3 1 1 3	-1

Замечание

В первом примере можно действовать так: воспользоваться такси первой фирмы и проехать маршрут $1 \to 3 \to 2$ за 2 единицы времени, затем выйти, вызвать за 6 единиц времени такси второй фирмы и проехать маршрут $2 \to 3 \to 4 \to 7$ за 37 единиц времени.

Во втором примере ни на одном перекрестке не работает интернет, а из отеля нельзя доехать до точки проведения олимпиады при помощи такси какой-либо одной фирмы.

Задача R. Скип или не скип

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

На улице уже 3024 год, идеи для задач давно закончились, а МКОШП теперь проходит в изменённом индивидуальном формате. Жюри выбирают n задач из всех, которые когда либо существовали, нумеруют их от 1 до n, присваивают каждой задаче стоимость a_i, а так же выбирают для каждой задачи некоторый параметр b_i от 1 до n.

Изначально тестирующая система выдаёт участнику **первую** задачу. Когда участнику выдают i-ю задачу, у него есть два варианта:

- либо он сдаёт задачу и получает a_i баллов.
- либо он может скипнуть задачу, тогда он никогда не сможет её сдать.

Далее тестирующая система выбирает для участника следующую задачу среди задач с номерами j, такими что:

- если он сдал задачу, она смотрит на задачи с номерами $j < i$.
- если он скипнул задачу, она смотрит на задачи с номерами $j \leqslant b_i$.

Среди этих задач она выбирает задачу с **максимальным** номером, которую она ранее **не выдавала** участнику (до этого он её не сдавал и не скипал). Если такой задачи нет, то соревнование для участника **завершается** и его результатом объявляется сумма баллов за все сданные задачи. В частности, если участник сдаёт первую задачу, то соревнование для него завершается.

Прохор очень хочет занять первое место, поэтому он тщательно готовился к этой олимпиаде, он прорешал все существующие задачи и теперь может сдать любую задачу, которую жюри могут выбрать на олимпиаду. Перед олимпиадой он задумался, сколько максимум баллов S он сможет получить в зависимости от выбора жюри.

К сожалению, Прохор очень устал от решения задач, помогите ему — для каждого выбора жюри ответьте, какой максимальный балл S сможет получить Прохор.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных — выборов жюри. В первой строке находится одно целое число t ($1 \leqslant t \leqslant 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leqslant n \leqslant 4 \cdot 10^5$) — число задач, взятых на олимпиаду.

Вторая строка каждого набора входных данных содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — стоимости задач.

Третья строка каждого набора входных данных содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant n$) — параметр задачи.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $4 \cdot 10^5$

Формат выходных данных

Для каждого набора входных данных выведите единственное число S — максимальное число баллов, которое может набрать Прохор.

Пример

стандартный ввод	стандартный вывод
2 2 15 15 2 1 4 100 200 300 400 3 4 4 1	15 700

Замечание

В первом случае Прохор может скипнуть первую задачу и система выдаст ему вторую задачу, он может сдать её и получить 15 баллов, далее не выданных задач не остается, соревнование для Прохора завершается с S = 15. Либо Прохор может сдать первую задачу, тоже получить 15 баллов и соревнование завершится.

Во втором случае Прохор скипает первую, переходит к третьей (она ранее не выдавалась), сдаёт её и получает 300 баллов, так как вторая задача ранее не выдавалась он переходит к ней, скипает её и переходит к четвёртой (она ранее не выдавалась), сдаёт четвёртую и получает ещё 400 баллов, не выданных задач не осталось, соревнование для Прохора завершается с S = 700.