

Задача А. 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 ($1 \leq n \leq 20000, 0 \leq m \leq 100000$). 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 ($1 \leq b_i, e_i \leq n, 0 \leq w_i \leq 100000$).

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 В. 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 $\text{НОД}(A, B) = M + 1 - i$.

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 ($1 \leq N, Q \leq 100000, 1 \leq M \leq N$) — 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 ответ — Q чисел, каждое в отдельной строке.

Пример

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

Задача С. 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 ($1 \leq N \leq 10^5, 1 \leq M \leq 3 \cdot 10^5, 1 \leq X \leq 10^9$).
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 . ($1 \leq w_i \leq 10^9$)

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 D. 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 ($1 \leq n \leq 10^5$). 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 ($1 \leq n \leq 10^5$) — число пользователей социальной сети, и m ($1 \leq m \leq 3 \cdot 10^5$) — число событий, произошедших за день. В следующих m строках содержится описание событий. Первое целое число в строке означает тип события t ($1 \leq t \leq 3$). Если $t = 1$, далее следует число i ($0 \leq i < n$), по которому можно вычислить, какой участник послал сообщение. Если $t = 2$, далее следуют числа i и j ($0 \leq i, j < n$), по которым можно вычислить номера участников, чаты с которыми должны объединиться. Если $t = 3$, далее следует число i ($0 \leq i < n$), 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 |
| 1 0 | 1 |
| 1 2 | 2 |
| 1 1 | |
| 1 2 | |
| 3 1 | |
| 2 1 2 | |
| 1 3 | |
| 3 3 | |
| 2 3 2 | |
| 3 2 | |

Задача Е. 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 ($1 \leq n \leq 5000$) — количество городов в Байтландии. Каждая из следующих n строк содержит по два целых числа x_i, y_i — координаты i -го города ($-10000 \leq x_i, y_i \leq 10000$). 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 F. Уничтожение массива

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

Вам дан массив, состоящий из n неотрицательных целых чисел a_1, a_2, \dots, a_n .

В этом массиве один за другим зачёркиваются числа. Вам задана перестановка чисел от 1 до n — порядок, в котором это происходит.

После зачёркивания очередного числа вам необходимо найти в этом массиве подотрезок с максимальной суммой, не содержащий ни одного зачёркнутого числа. Сумму чисел в пустом подотрезке считайте равной 0.

Формат входных данных

В первой строке входных данных записано число n ($1 \leq n \leq 100\,000$) — длина массива.

В второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$).

В третьей строке входных данных записана перестановка чисел от 1 до n — порядок, в котором зачеркиваются числа.

Формат выходных данных

В выходной файл выведите n строк, каждая из которых должна содержать одно число — максимальную сумму на подотрезке заданного массива, не содержащем зачёркнутых чисел, после выполнения очередного действия.

Примеры

| стандартный ввод | стандартный вывод |
|---|---|
| 4 1 3 2 5 3 4 1 2 | 5 4 3 0 |
| 5 1 2 3 4 5 4 2 3 5 1 | 6 5 5 1 0 |
| 8 5 5 4 4 6 6 5 5 5 2 8 7 1 3 4 6 | 18 16 11 8 8 6 6 0 |

Замечание

В первом тестовом примере происходит следующее:

- Зачеркивается третий элемент, массив принимает вид $1\ 3\ *\ 5$. Отрезок с максимальной суммой 5 состоит из одного числа 5.
- Зачеркивается четвертый элемент, массив принимает вид $1\ 3\ *\ *$. Отрезок с максимальной суммой 4 состоит из двух чисел 1 3.
- Зачеркивается первый элемент, массив принимает вид $*\ 3\ *\ *$. Отрезок с максимальной суммой 3 состоит из одного числа 3.

4. Зачеркивается оставшийся второй элемент, в этот момент непустых допустимых подотрезков не остается, поэтому здесь ответ равен нулю.

Задача G. Дороги

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

В 2179 году Tinkoff.Generation планирует провести летнюю школу на территории одного из архипелагов в Тихом океане. Но есть одна проблема: сейчас не с каждого острова можно попасть на каждый. А именно, между некоторыми парами островов проложены мосты, но при этом может быть так, что с одного острова на другой нельзя попасть, проходя по мостам с острова на остров.

Ваша задача заключается в том, чтобы исправить это, чтобы школьники, живущие на разных островах, могли ходить друг к другу в гости.

Поскольку постройка мостов — дело недешёвое, вам нужно потратить как можно меньше денег. Для простоты мы будем считать, что острова задаются точками на плоскости, а стоимость постройки моста между островами равняется евклидову расстоянию между этими точками.

Формат входных данных

В первой строке входных данных содержится одно целое число N ($1 \leq N \leq 50$) — количество островов в архипелаге. Далее в N строках записано по 2 натуральных числа — координаты островов (координаты не превышают 100). Гарантируется, что все точки попарно различны. Далее дано число M — количество уже соединённых мостов пар точек. В следующих M строках записаны по 2 числа — номера островов, соединённых очередным мостом. Острова нумеруются с единицы.

Формат выходных данных

Вывести единственное число — минимальный возможный бюджет, который надо запланировать для того, чтобы соединить все острова в одну компоненту связности. Ваш ответ может отличаться от правильного не раньше, чем в 6 знаке после запятой.

Примеры

| стандартный ввод | стандартный вывод |
|-----------------------------|-------------------|
| 1 1 1 0 | 0.000000 |
| 2 1 1 2 2 0 | 1.414214 |
| 2 1 1 2 2 1 1 2 | 0.000000 |

Задача Н. Разрезание графа

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

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 ($1 \leq n \leq 50\,000$, $0 \leq m \leq 100\,000$, $m \leq k \leq 150\,000$).

Следующие m строк задают рёбра графа; i -я из этих строк содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$), 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`» ($1 \leq u, v \leq n$), которая означает, что из графа удаляют ребро между вершинами u и v . Операция типа `ask` задаётся строкой «`ask u v`» ($1 \leq u, v \leq n$), 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 | YES |
| 1 2 | YES |
| 2 3 | NO |
| 3 1 | NO |
| ask 3 3 | |
| cut 1 2 | |
| ask 1 2 | |
| cut 1 3 | |
| ask 2 1 | |
| cut 2 3 | |
| ask 3 1 | |

Задача I. Сделай связным

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

Вам дан неориентированный граф, состоящий из n вершин. На каждой вершине записано число; число, записанное на вершине i , равно a_i . Изначально в графе нет ни одного ребра.

Вы можете добавлять ребра в граф за определенную стоимость. За добавление ребра между вершинами x и y надо заплатить $a_x + a_y$ монет. Также существует m специальных предложений, каждое из которых характеризуется тремя числами x , y и w , и означает, что можно добавить ребро между вершинами x и y за w монет. Эти специальные предложения не обязательно использовать: если существует такая пара вершин x и y , такая, что для нее существует специальное предложение, можно все равно добавить ребро между ними за $a_x + a_y$ монет.

Сколько монет минимально вам потребуется, чтобы сделать граф связным? Граф является связным, если от каждой вершины можно добраться до любой другой вершины, используя только ребра этого графа.

Формат входных данных

В первой строке заданы два целых числа n и m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$) — количество вершин в графе и специальных предложений, соответственно.

Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{12}$) — числа, записанные на вершинах.

Затем следуют m строк, в каждой из которых заданы три целых числа x , y и w ($1 \leq x, y \leq n$, $1 \leq w \leq 10^{12}$, $x \neq y$), обозначающие спецпредложение: можно добавить ребро между вершинами x и y за w монет.

Формат выходных данных

Выведите одно целое число — минимальное количество монет, которое необходимо потратить, чтобы сделать граф связным.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 3 2 1 3 3 2 3 5 2 1 1 | 5 |
| 4 0 1 3 3 7 | 16 |
| 5 4 1 2 3 4 5 1 2 8 1 3 10 1 4 7 1 5 15 | 18 |

Замечание

В первом примере из условия можно соединить 1 и 2 при помощи 2-го спецпредложения, а затем 1 и 3 без использования спецпредложения.

В следующих двух примерах оптимальный ответ можно получить без использования спецпредложений.

Задача J. 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 экономичных схем альтернативного электроснабжения школ.

Формат входных данных

В первой строке входного файла находятся два натуральных числа, разделенных пробелом: N ($3 \leq N \leq 100$), 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 ($1 \leq C_i \leq 300$) 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 ($S_1 \leq S_2$). $S_1 = S_2$ 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 К. Плотное остовное дерево

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

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

Формат входных данных

Первая строка содержит два целых числа n и m ($n \leq 1000$, $m \leq 10000$) — количество вершин и ребер в графе.

Каждая из следующих m строк содержит три целых числа b_i , e_i и w_i ($1 \leq b_i, e_i \leq n$, $0 \leq |w_i| \leq 10^9$) — номера вершин, которые соединяет ребро, а также вес ребра.

Формат выходных данных

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

В противном случае в единственной строке выведите NO.

Пример

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2 | YES 0 |