

Задача А. Остовное дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
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. 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 С. День Объединения

Имя входного файла: `unionday.in`
Имя выходного файла: `unionday.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

64 мегабайта

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 ($1 \leq n \leq 5\,000$) — количество городов в Байтландии. Каждая из следующих n строк содержит по два целых числа x_i, y_i — координаты i -го города ($-10\,000 \leq x_i, y_i \leq 10\,000$). 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

| <code>unionday.in</code> | <code>unionday.out</code> |
|--------------------------|---------------------------|
| 6 | 9.6568542495 |
| 1 1 | |
| 7 1 | |
| 2 2 | |
| 6 2 | |
| 1 3 | |
| 7 3 | |

Задача D. Разрезание графа

Имя входного файла: `cutting.in`
Имя выходного файла: `cutting.out`
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

| <code>cutting.in</code> | <code>cutting.out</code> |
|-------------------------|--------------------------|
| 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 | |