

Задача А. Минимальное остовное дерево

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан связный неориентированный взвешенный граф. Необходимо выбрать в этом графе некоторое подмножество рёбер минимального суммарного веса таким образом, чтобы между любыми двумя вершинами графа существовал путь из выбранных рёбер.

Очевидно, что выбранное подмножество рёбер должно быть деревом (для минимальности суммарного веса ребер), и такое подмножество называется *минимальным остовным деревом* или *минимальным каркасом*.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 20\,000$, $0 \leq m \leq 100\,000$). Следующие m строк содержат описание рёбер 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 ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Гарантируется, что данный граф является связным.

Формат выходных данных

Программа должна вывести одно целое число — вес минимального остовного дерева.

Примеры

<code>stdin</code>	<code>stdout</code>
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Задача В. Разрезание графа

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 3 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 ($1 \leq n \leq 50\,000$, $0 \leq m \leq 100\,000$, $m \leq k \leq 150\,000$).

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

stdin	stdout
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 С. Нефтяное дело

Имя входного файла: oil.in
Имя выходного файла: oil.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дан неориентированный связный граф из n вершин и m рёбер. Для каждого ребра известна стоимость его удаления в тугриках. У вас есть s тугриков. Вы хотите удалить как можно больше рёбер так, чтобы граф оставался связным, а суммарная стоимость всех удалённых рёбер не превосходила s тугриков.

Формат входных данных

Первая строка входных данных содержит три целых числа n , m и s — количество вершин в графе, количество рёбер в графе и количество тугриков соответственно ($2 \leq n \leq 50\,000$, $1 \leq m \leq 100\,000$, $0 \leq s \leq 10^{18}$). Следующие m строк содержат описания рёбер графа. Каждое описание состоит из трёх целых чисел — номера вершин, которые соединяет данное ребро, и стоимость удаления ребра в тугриках (стоимость не превосходит 10^9). В графе не бывает кратных рёбер и петель.

Формат выходных данных

В первой строке выведите максимальное количество удаляемых рёбер. Во второй строке выведите номера удаляемых рёбер (рёбра нумеруются с единицы в порядке, данном во входном файле).

Примеры

oil.in	oil.out
6 7 10	2
1 2 3	1 6
1 3 3	
2 3 3	
3 4 1	
4 5 5	
5 6 4	
4 6 5	