

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. День Объединения

Имя входного файла: `unionday.in`
Имя выходного файла: `unionday.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 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 ($1 \leq n \leq 5000$) — 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^{-3} .

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. Горилла и множества

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

Однажды молодая горилла нашла n множеств. В множестве i ($1 \leq i \leq n$) находится единственное число i . Андрей Сергеевич нашел эту молодую гориллу и придумал ей полезное занятие. Требуется ответить на m запросов:

1. Объединить множества, в которых находятся числа a и b , если они находятся в разных множествах.
2. Сказать, находятся ли в одном множестве числа a и b .
3. Сказать, в скольки объединениях уже участвовало число a .

Формат входных данных

В первой строке через пробел даны два целых числа n, m ($1 \leq n, m \leq 2 \times 10^5$). В следующих m строках даны запросы. Сначала вводится целое число $type$ ($1 \leq type \leq 3$). Если $type \in \{1, 2\}$, то через пробел вводятся целые числа a, b ($1 \leq a, b \leq n$), иначе вводится целое число a ($1 \leq a \leq n$).

Формат выходных данных

На каждый запрос с $type = 2$ выведите в отдельной строке YES, если a и b лежат в одном множестве, иначе NO. На каждый запрос с $type = 3$ выведите в отдельной строке целое число - количество объединений, в которых участвовало число a .

Пример

стандартный ввод	стандартный вывод
5 8	NO
2 3 1	0
3 3	YES
1 2 4	1
2 1 1	NO
3 4	2
2 3 4	
1 3 4	
3 4	

Задача Е. 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	

Задача F. Уничтожение массива

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

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