

Вес компоненты

Имя входного файла: `components.in`
Имя выходного файла: `components.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

В неориентированный взвешенный граф добавляют ребра. Напишите программу, которая в некоторые моменты находит сумму весов ребер в компоненте связности.

Формат входного файла

В первой строке записано два числа n, m ($1 \leq n, m \leq 10^6$) — количество вершин в графе и количество производимых добавлений и запросов. Далее следует m строк с описанием добавления или запроса. Каждая строка состоит из двух или четырех чисел. Первое из чисел обозначает код операции. Если первое число 1, то за ним следует еще три числа x, y, w . Это означает, что в граф добавляется ребро из вершины x в вершину y веса w . ($1 \leq x < y \leq n, 1 \leq w \leq 10^3$). Кратные ребра допустимы. Если первое число 2, то за ним следует ровно одно число x . Это означает, что необходимо ответить на вопрос, какова сумма ребер в компоненте связности, которой принадлежит вершина x ($1 \leq x \leq n$).

Формат выходного файла

Для каждой операции с кодом 2 выведите ответ на поставленную задачу. Ответ на каждый запрос выводите на отдельной строке.

Примеры

<code>components.in</code>	<code>components.out</code>
6 10	0
2 1	1
1 1 2 1	3
2 1	6
1 2 4 2	3
2 1	0
1 1 4 3	
2 1	
1 3 5 3	
2 5	
2 6	

Остовное дерево 2

Имя входного файла: `spantree2.in`
Имя выходного файла: `spantree2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Требуется найти в связном графе остовное дерево минимального веса.

Формат входного файла

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

Граф является связным.

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

Примеры

<code>spantree2.in</code>	<code>spantree2.out</code>
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Хорошие дни

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

Билл разрабатывает новую математическую теорию, описывающую человеческие эмоции. Его последние исследования посвящены изучению того, насколько хорошие и плохие дни влияют на воспоминания людей о различных периодах жизни.

Недавно Билл придумал методику, которая описывает, насколько был хорошим или плохим день человеческой жизни с помощью сопоставления дню некоторого неотрицательного целого числа. Билл называет это число *эмоциональной значимостью* этого дня. Чем больше это число, тем лучше этот день. Билл полагает, что значимость некоторого периода человеческой жизни равна сумме эмоциональных значимостей каждого из дней периода, помноженной на минимум эмоциональных значимостей дней этого периода. Эта методика отражает то, что период, который в среднем может быть весьма неплох, бывает испорчен одним плохим днем.

Теперь Билл хочет проанализировать свою собственную жизнь и найти в ней период максимальной значимости. Помогите ему это сделать.

Формат входного файла

Первая строка входного файла содержит число n — количество дней в жизни Билла, которые он хочет исследовать ($1 \leq n \leq 100\,000$). Оставшаяся часть файла содержит n целых чисел a_1, a_2, \dots, a_n , все в пределах от 0 до 10^6 — эмоциональные значимости дней. Числа во входном файле разделяются пробелами и переводами строки.

Формат выходного файла

В первой строке выходного файла выведите максимальную значимость периода жизни Билла. Во второй строке выведите два числа l и r , обозначающие, что значимость периода с l -го по r -й день (включительно) в жизни Билла была максимально возможной.

Пример

<code>feelgood.in</code>	<code>feelgood.out</code>
6	60
3 1 6	3 5
4 5 2	

Unionday. 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 решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было бы 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 входного файла

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 5\,000$) — количество городов в Байтландии. Каждая из следующих n строк содержит 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 выходного файла

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее 10^{-3} .

Примеры

<code>unionday.in</code>	<code>unionday.out</code>
6	9.65685
1 1	
7 1	
2 2	
6 2	
1 3	
7 3	