

Задача А. Остовное дерево (1 балл)

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

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \leq n \leq 5000$). Каждая из следующих n строк содержит два целых числа x_i, y_i — координаты i -й вершины ($-10\,000 \leq x_i, y_i \leq 10\,000$). Никакие две точки не совпадают.

Формат выходного файла

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

Примеры

spantree.in	spantree.out
3 0 0 1 0 0 1	2

Задача В. Остовное дерево 2 (1 балл)

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

spantree2.in	spantree2.out
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Задача С. Плотное остовное дерево (1 балл)

Имя входного файла: `mindiff.in`
Имя выходного файла: `mindiff.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

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

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 ($1 \leq b_i, e_i \leq n$, $0 \leq |w_i| \leq 10^9$). $n \leq 1000$, $m \leq 10\,000$.

Формат выходного файла

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

В противном случае в единственной строке выведите NO.

Примеры

mindiff.in	mindiff.out
4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2	YES 0

Задача D. Эйлеров путь (1 балл)

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

Дан неориентированный связный граф, не более трех вершин имеет нечетную степень. Требуется определить, существует ли в нем путь, проходящий по всем ребрам.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \leq n \leq 1000$). Далее следуют n строк, задающих ребра. В i -ой из этих строк находится число m_i — количество ребер, инцидентных вершине i . Далее следуют m_i натуральных чисел — номера вершин, в которые идет j -ое ребро из i -ой вершины.

Граф может содержать кратные ребра, но не содержит петель.

Формат выходного файла

Если решение существует, то в первую строку выходного файла выведите одно число k — количество ребер в искомом маршруте, а во вторую $k + 1$ число — номера вершин в порядке их посещения.

Если решений нет, выведите в выходной файл одно число -1.

Если решений несколько, выведите любое.

Примеры

euler.in	euler.out
4	5
2 2 2	1 2 3 4 2 1
4 1 4 3 1	
2 2 4	
2 3 2	

Задача Е. Алгоритм двух китайцев (3 балла)

Имя входного файла: `chinese.in`
Имя выходного файла: `chinese.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный взвешенный граф. Покрывающим деревом с корнем в вершине u назовем множество ребер, таких что из вершины u достижима любая другая вершина v , притом единственным образом. Весом дерева назовем сумму весов его ребер.

Требуется определить, существует ли в данном графе покрывающее дерево с корнем в вершине с номером 1. В случае существования требуется определить его минимальный вес.

Формат входного файла

В первой строке входного файла два числа: n и m ($2 \leq n \leq 1000, 1 \leq m \leq 10000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее по модулю 10^9 .

Формат выходного файла

Если покрывающее дерево существует, выведите в первой строке выходного файла YES, а во второй строке целое число — его минимальный вес. В противном случае в единственной строке выведите NO.

Примеры

<code>chinese.in</code>	<code>chinese.out</code>
2 1 2 1 10	NO
4 5 1 2 2 1 3 3 1 4 3 2 3 2 2 4 2	YES 6