

Содержание

0-1 Bfs	2
Задача А. Island. Островные государства [0.2 sec, 256 mb]	2
Spfa	3
Задача В. Путь в случайном графе [1 sec, 256 mb]	3
Floyd	4
Задача С. Цикл отрицательного веса [1 sec, 256 mb]	4
Dijkstra	5
Задача D. Расстояние между вершинами [1 sec, 256 mb]	5
Задача Е. Расстояние между вершинами [1 sec, 256 mb]	6
Ford-Bellman	7
Задача F. Path. Кратчайший путь [1 sec, 256 mb]	7
Suprise!	8
Задача G. Сумма расстояний v2.0 [4 sec, 256 mb]	8

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

Обратите внимание, что ввод-вывод во всех задачах стандартный.

0-1 Bfs

Задача А. Island. Островные государства [0.2 сек, 256 mb]

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город N на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

Формат входных данных

В первой строке входного файла записано два числа N и M ($1 \leq N, M \leq 100\,000$) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах — N чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних M строках записаны пары $1 \leq a, b \leq N, a \neq b$. Каждая пара означает наличие дороги из города a в город b . По дорогам Байтландии можно двигаться в любом направлении.

Формат выходных данных

Если искомого пути не существует, выведите единственное слово `impossible`. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

Пример

stdin	stdout
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	1 3 1 4 5

Spfa

Задача В. Путь в случайном графе [1 сек, 256 mb]

Дан случайный взвешенный ориентированный граф из n вершин, m рёбер. Гарантируется, что в графе нет отрицательных циклов. Ваша задача – найти расстояние от вершины s до вершины t . Что такое случайный граф? Рёбра равномерно случайного веса между равномерно случайными вершинами.

Формат входных данных

На первой строке $n\ m\ s\ t$ ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$, $1 \leq s, t \leq n$). На следующих m строках тройки чисел $a_i\ b_i\ w_i$ ($1 \leq a, b \leq n$, $|w| < 10^9$), рёбра из a_i в b_i веса w_i .

Формат выходных данных

Одно число – длину кратчайшего пути.

Если пути из s в t не существует, выведите число 10^{18} .

Примеры

stdin	stdout
2 0 1 2	1000000000000000000
3 4 1 3 1 2 20 2 3 -15 1 3 10 3 1 -4	5

Floyd

Задача С. Цикл отрицательного веса [1 sec, 256 mb]

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

Формат входных данных

Во входном файле в первой строке число N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках находится по N чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае.

Пример

stdin	stdout
2 0 -1 -1 0	YES

Dijkstra

Задача D. Расстояние между вершинами [1 sec, 256 mb]

Дан неориентированный взвешенный граф. Найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит натуральные числа N , M , вторая строка содержит натуральные числа S и F ($N \leq 5\,000$, $M \leq 100\,000$, $1 \leq S, F \leq N$, $S \neq F$) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие M строк по три натуральных числа b_i , e_i и w_i — номера концов i -ого ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами S и F . Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода.

Если путь из S в F не существует, выведите -1 .

Пример

stdin	stdout
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3 1 2 3

Задача Е. Расстояние между вершинами [1 сек, 256 mb]

Коль Дейкстру́ писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей
ЛКШ.июль-2007

Дан взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leq s, t \leq n$, $s \neq t$).

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 ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100$).

$n \leq 100\,000$, $m \leq 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t .

Если путь из s в t не существует, выведите -1.

Пример

stdin	stdout
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3

Ford-Bellman

Задача F. Path. Кратчайший путь [1 sec, 256 mb]

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u .

Формат входных данных

Первая строка входного файла содержит n , m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leq n \leq 2\,000$, $1 \leq m \leq 5\,000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходных данных

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u , '*' если не существует путь из s в u и '-' если не существует кратчайший путь из s в u .

Пример

stdin	stdout
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

Suprise!

Задача G. Сумма расстояний v2.0 [4 сек, 256 mb]

Дан невзвешенный ориентированный граф. Определим $f(u, v)$ как длину кратчайшего пути между вершинами u и v (если пути не существует, скажем, что $f(u, v) = 0$).

Найдите $\sum_{u=1}^n \sum_{v=1}^n f(u, v)$.

Формат входных данных

В первой строке содержится число n — размер графа ($2 \leq n \leq 2000$).

В каждой из последующих n строк задано по n чисел. j -е число в i -й строке равняется 1, если существует ориентированное ребро из вершины i в вершину j .

Формат выходных данных

Выведите одно число — сумму попарных кратчайших путей.

Примеры

stdin	stdout
3 0 1 1 1 0 1 1 1 0	6
4 0 1 1 1 0 0 1 1 0 0 0 1 0 0 0 0	6
3 0 1 1 0 0 0 0 0 0	2