

Задача А. Расстояние между вершинами

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

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

Спектакль преподавателей ЛКШ.июль-2007

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

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$). 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 рёбер по одному на строке. Рёбро номер i описывается тремя натуральными числами b_i , e_i и w_i — номерами концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100$).

Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t , или -1, если такого пути нет.

Пример

| distance.in | distance.out |
|--|--------------|
| 4 4 1 3 1 2 1 2 3 2 3 4 5 4 1 4 | 3 |

Задача В. Pink Floyd

Имя входного файла: `floyd.in`
 Имя выходного файла: `floyd.out`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

Группа *Pink Floyd* собирается отправиться в новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист *Роджер Уотерс* постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным.

Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

Формат входного файла

Первая строка входного файла содержит три натуральных числа n , m и k — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ($n \leq 100$, $m \leq 10\,000$, $2 \leq k \leq 10\,000$). Города пронумерованы числами от 1 до n .

Следующие m строк содержат описание рейсов, по одному на строке. Рейс номер i описывается тремя числами b_i , e_i и

w_i — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ($1 \leq b_i, e_i \leq n$, $-100\,000 \leq w_i \leq 100\,000$).

Последняя строка содержит числа a_1, a_2, \dots, a_k — номера городов, в которых проводятся концерты ($a_i \neq a_{i+1}$). В начале концертного тура группа находится в городе a_1 .

Гарантируется, что группа может дать все концерты.

Формат выходного файла

Первая строка выходного файла должна содержать число l — количество рейсов, которые должна сделать группа. Вторая строка должна содержать l чисел — номера используемых рейсов.

Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку `"infinitely kind"`.

Примеры

| floyd.in | floyd.out |
|--|------------------|
| 4 8 5 1 2 -2 2 3 3 3 4 -5 4 1 3 1 3 2 3 1 -2 3 2 -3 2 4 -10 1 3 1 2 4 | 6 5 6 5 7 2 3 |
| 4 8 5 1 2 -2 2 3 3 3 4 -5 4 1 3 1 3 2 3 1 -2 3 2 -3 2 4 10 1 3 1 2 4 | infinitely kind |

Задача С. Островные государства — 2

Имя входного файла: `island2.in`
 Имя выходного файла: `island2.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

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. Кроме этого, за выезд из городов с чётными номерами берется удвоенная пошлина.

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 входного файла

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

В следующий 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 выходного файла

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

Примеры

| island2.in | island2.out |
|--|------------------|
| 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 | 2 3 1 4 5 |

Задача D. 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 входного файла

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 ($-10\,000 \leq x_i, y_i \leq 10\,000$). 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} .

Пример

| unionday.in | unionday.out |
|---|--------------|
| 6 1 1 7 1 2 2 6 2 1 3 7 3 | 9.6568542495 |

Задача E. Кратчайшие пути

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

Вам дан взвешенный ориентированный граф и вершина s в нём. Для каждой вершины графа u выведите длину кратчайшего пути от вершины s до вершины u .

Формат входного файла

Первая строка входного файла содержит три целых числа n, m, s — количество вершин и ребёр в графе и номер начальной вершины соответственно ($2 \leq n \leq 2000$, $1 \leq m \leq 5000$).

Следующие m строчек описывают рёбра графа. Каждое ребро задаётся тремя числами — начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра — целое число, не превосходящее 10^{15} по абсолютной величине. В графе могут быть кратные рёбра и петли.

Формат выходного файла

Выведите n строчек — для каждой вершины u выведите длину кратчайшего пути из s в u . Если не существует пути между s и u , выведите «*». Если не существует кратчайшего пути между s и u , выведите «-».

Примеры

| path.in | path.out |
|---|---------------------------------------|
| 6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1 | 0 10 - - - * - - |