

Содержание

Must have	2
Задача 5А. Стоимость проезда [0.1 sec, 256 mb]	2
Задача 5В. Расстояние между вершинами [0.5 sec, 256 mb]	3
Обязательные задачи	4
Задача 5С. Сумма расстояний [0.2 sec, 256 mb]	4
Задача 5D. Кратчайший путь двух коней [0.1 sec, 256 mb]	5
Задача 5Е. Расстояние между вершинами [1 sec, 256 mb]	6
Дополнительные задачи	7
Задача 5F. Кратчайший путь [1 sec, 256 mb]	7
Задача 5G. Грязь [0.15 sec, 256 mb]	8
Задача 5H. Идеальный путь [1 sec, 256 mb]	9

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

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

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Must have

Задача 5А. Стоимость проезда [0.1 sec, 256 mb]

Страна состоит из n городов и m дорог. Города пронумерованы числами от 1 до n . Город с номером s является столицей. Все дороги односторонние, проход по каждой дороге стоит ровно 1 золотой. Требуется найти минимальные стоимости проезда от каждого города до столицы.

Формат входных данных

В первой строке файла записаны три целых числа — n , s и m (количество городов, номер столичного города и количество дорог).

В следующих m строках записаны пары чисел. Пара чисел (a, b) означает, что есть дорога из города a в город b .

Ограничения: $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$.

Формат выходных данных

Выведите n чисел — минимальные стоимости проезда от городов до столицы. Если от какого-то города не существует ни одного пути до столицы, выведите -1 .

Пример

stdin	stdout
3 2 2	1 0 -1
1 2	
2 3	

Задача 5В. Расстояние между вершинами [0.5 sec, 256 mb]

Дан взвешенный неориентированный 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

Обязательные задачи

Задача 5С. Сумма расстояний [0.2 sec, 256 mb]

Дан связный неориентированный граф. Требуется найти сумму расстояний между всеми парами вершин.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 1000$, $0 \leq m \leq 10\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Гарантируется, что граф связан.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — сумму попарных расстояний между вершинами.

Пример

stdin	stdout
5 5 1 2 2 3 3 4 5 3 1 5	16

Замечание

Не всё, что матрица, то Флойд.

Задача 5D. Кратчайший путь двух коней [0.1 sec, 256 mb]

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке.

Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений.

Пример

stdin	stdout
a1	1 b3
c2	1 d4
c2	2 a1
a1	1 c2

Замечание

Задача специально для тех, кто боится строить графы.

Задача 5Е. Расстояние между вершинами [1 sec, 256 mb]

Дан неориентированный взвешенный граф без петель и кратных рёбер. 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 рёбер 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	3
1 3	1 2 3
1 2 1	
3 4 5	
3 2 2	
4 1 4	

Замечание

Существует жизнь без кучи.

Дополнительные задачи

Задача 5F. Кратчайший путь [1 sec, 256 mb]

Надеюсь, все вы умеете искать в ориентированном графе кратчайший путь. В этой задаче вам предлагается свое умение продемонстрировать.

Вам дан ориентированный взвешенный граф. Веса ребер — целые числа от 1000 до 2000. Нужно несколько раз (не более 1000) ответить на следующий запрос: длина кратчайшего пути из некоторой вершины s в некоторую вершину t .

Формат входных данных

На первой строке числа N и M ($1 \leq N \leq 25\,000$, $0 \leq M \leq 50\,000$) — количество вершин и ребер нашего графа, соответственно. Вершины нумеруются целыми числами от 1 до N . Далее M строк содержат информацию о ребрах графа. Каждое ребро задается тремя числами — номер начала, номер конца и вес. Все веса — целые числа от 1000 до 2000. В графе могут быть и петли, и кратные ребра. Следующая строка содержит число K ($1 \leq K \leq 1000$) — количество запросов. В следующих K строках задаются запросы. Каждый запрос описывается двумя числами — из какой вершины, и в какую должен вести путь.

Формат выходных данных

Для каждого запроса выведите на отдельной строке целое число — длину кратчайшего пути. Если кратчайшего пути не существует следует вывести -1 .

Пример

stdin	stdout
5 5	-1
1 2 2000	3000
1 3 1000	0
1 4 1200	2000
2 3 1500	
3 4 1500	
4	
1 5	
2 4	
3 3	
1 2	

Замечание

Путем в графе называется такая последовательность ребер, что конец i -го совпадает с началом $i+1$ -го. Длиной пути называется суммарный вес ребер. Путь является кратчайшим, если его длина минимальна.

Подсказка по решению

Вы можете поихать Дейкстру или Форд-Беллмана, но есть решение получше.

Задача 5G. Грязь [0.15 sec, 256 mb]

— Здравствуйте! Могу я поговорить с Петровым? Алё, милый, привет... ты знаешь, у нас дома небольшая авария произошла... Но твой компьютер не пострадал, не волнуйся. Но теперь там немного грязно. Ну, то есть очень грязно. Но ты не волнуйся, я приготовила тебе твои болотные сапоги, у входа стоят. А грязь я уберу, как будет свободное время. Когда? Ну, наверное, когда в отпуск пойду. А, ну когда вернёмся из Турции. А, ну значит в следующий отпуск, но обязательно уберу. А пока я у мамы поживу. И ты, кстати, тоже можешь. Ну, как хочешь, я же не заставляю... Только пока я не убрала, ты там грязь не разводи, сильно сапогами по грязи не шлёпай и когда по чистому ходишь, сапоги снимай и тапочки обувай, я их тоже возле входа поставила, ты их бери с собой, когда идёшь по грязи и переобувай. А когда по чистому идёшь, бери сапоги, там грязь в разных местах. Программисты, как известно, не самые трудолюбивые люди, поэтому убирать грязь не станут. Но переобувать болотные сапоги каждый раз, когда переходишь от грязного пола к чистому и наоборот — удовольствие ниже среднего, уж лучше пройти лишние несколько метров. Чтобы прожить время до следующего отпуска с комфортом, надо срочно выработать способ добираться из одной точки квартиры с минимальным количеством переобуваний по пути, ну а уж среди них, конечно, выбрать самый короткий.

Формат входных данных

В первой строке даны два целых числа M и N — размеры квартиры (в у.е.). $1 \leq N, M \leq 500$. Два целых числа во второй строке — координаты компьютера (в у.е.), а два целых числа в третьей строке — координаты холодильника (тоже в у.е.). Далее идут M строк по N символов в каждой — план квартиры. На плане 1 означает чистое место, 2 — грязное, 0 — стена или непроходимая грязь. Переходить можно только на клетки, имеющие общую вершину с данной, при переходе с чистой на грязную и наоборот надо переобуваться. Холодильник и компьютер находятся не в клетках, помеченных нулём. Левая верхняя клетка плана имеет координаты (1, 1).

Формат выходных данных

Длина кратчайшего пути (количество преодоленных квадратиков, включая начальный и конечный) с минимальным количеством переобуваний, и, через пробел, количество переобуваний (переобувание проходит при переходе с грязного на чистое и наоборот). Если пройти к холодильнику невозможно, вывести числа 0 0.

Пример

stdin	stdout
3 7 1 1 3 7 1200121 1212020 1112021	8 4

Замечание

Подразумевается линейное решение.

К сожалению, очень хорошо написанная Дейкстра тоже получит ОК.

Задача 5Н. Идеальный путь [1 sec, 256 mb]

Есть лабиринт развлечений из n комнат и m цветных переходов между комнатами. Участник изначально попадает в комнату 1, цель — добраться до комнаты n . Несколько участников одновременно стартуют из комнаты 1. Каждый участник пройдет некоторым путем из комнаты 1 в комнату n , записывая цвета переходов, по которым прошел. Выиграет участник с кратчайшей последовательностью, а если таких несколько, с лексикографически меньшей среди кратчайших. Эндрю тоже участвует и очень хочет выиграть. Помогите ему, найдите кратчайшую, а из таких лексикографически минимальную последовательность переходов из 1 в n .

Замечание

Последовательность (a_1, a_2, \dots, a_k) лексикографически меньше последовательности (b_1, b_2, \dots, b_k) если существует i такое, что $a_i < b_i$ и $a_j = b_j$ для всех $j < i$.

Формат входных данных

Первая строка содержит целые числа n и m — количество комнат и переходов соответственно ($2 \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$). Следующие m строк содержат описания переходов. Каждый переход задается тремя целыми числами: a_i , b_i и c_i — номера комнат, которые он соединяет, и цвет перехода ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^9$). Каждый переход может быть использован в обоих направлениях. Две комнаты могут быть соединены несколькими переходами, также может быть переход из комнаты в саму себя. Гарантируется, что из комнаты 1 можно как-нибудь попасть в комнату n .

Формат выходных данных

На первой строке выведите k — длину кратчайшей последовательности переходов из комнаты 1 в комнату n . На следующей строке k чисел — цвета переходов в порядке их прохождения.

Пример

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