

## Содержание

<b>Задачи здорового человека</b>	<b>2</b>
<b>Задача 5А. Сумма расстояний [0.2 sec, 256 mb]</b>	<b>2</b>
<b>Задача 5В. Расстояние между вершинами [0.5 sec, 256 mb]</b>	<b>3</b>
<b>Для искателей острых ощущений</b>	<b>4</b>
<b>Задача 5С. Стоимость проезда [0.1 sec, 256 mb]</b>	<b>4</b>
<b>Задача 5D. Longpath. Длиннейший путь [0.1 sec, 256 mb]</b>	<b>5</b>
<b>Задача 5Е. Расстояние между вершинами [0.2 sec, 256 mb]</b>	<b>6</b>

---

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**.

---

## Задачи здорового человека

### Задача 5А. 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

Это не про Флойда. Это не про Дейкстру.

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

$1 \leq n \leq 100\,000, 1 \leq m \leq 200\,000$ .

#### 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

#### Замечание

Дейкстра обыкновенная. `set<pair<int,int>` зайдет. `priority_queue` быстрее.

Эстетсы вместо пар могут создать `struct Vertex {int id;}` и перегрузить `operator<`.

## Для искателей острых ощущений

### Задача 5С. Стоимость проезда [0.1 сек, 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	

#### Замечание

Казалось бы bfs. Или нет? Или всё-таки да?

### Задача 5D. Longpath. Длиннейший путь [0.1 sec, 256 mb]

Дан ориентированный граф без циклов. Требуется найти в нем длиннейший путь.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и дуг графа соответственно. Следующие  $m$  строк содержат описания дуг по одной на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i$  и  $e_i$  — началом и концом дуги соответственно ( $1 \leq b_i, e_i \leq n$ ).

Входной граф не содержит циклов и петель.

$n \leq 10\,000$ ,  $m \leq 100\,000$ .

#### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — количество дуг в длиннейшем пути.

#### Пример

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

#### Замечание

В орграфе без циклов мы умеем почти что угодно считать dfs-ом.

### Задача 5Е. Расстояние между вершинами [0.2 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.

Матрица смежности в этой задаче крайне не эффективна из-за проблем с кешом.