

## Задача А. Расстояние между вершинами

Имя входного файла: `distance.in`  
Имя выходного файла: `distance.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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 входных данных

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. Ребро 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 выходных данных

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

### Пример

<code>distance.in</code>	<code>distance.out</code>
4 4 1 3 1 2 1 2 3 2 3 4 5 4 1 4	3

## Задача В. Эвакуация

Имя входного файла: `evacuation.in`  
Имя выходного файла: `evacuation.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Одна из Сверхсекретных организаций, чье название мы не имеем право разглашать, представляет собой сеть из  $N$  подземных бункеров, соединенных равными по длине туннелями, по которым из любого бункера можно добраться до любого другого (не обязательно напрямую). Связь с внешним миром осуществляется через специальные засекреченные выходы, которые расположены в некоторых из бункеров. Организации понадобилось составить план эвакуации персонала на случай экстренной ситуации. Для этого для каждого из бункеров необходимо узнать, сколько времени потребуется для того, чтобы добраться до ближайшего из выходов. Вам, как специалисту по таким задачам, поручено рассчитать необходимое время для каждого из бункеров по заданному описанию помещения Сверхсекретной организации. Для вашего же удобства бункеры занумерованы числами от 1 до  $N$ .

### Формат входных данных

В первой строке записано число  $N$ , во второй — число  $K$  ( $1 \leq N \leq 100\,000$ ,  $1 \leq K \leq N$ ) — количество бункеров и количество выходов соответственно. Далее через пробел записаны  $K$  различных чисел от 1 до  $N$ , обозначающих номера бункеров, в которых расположены выходы. Потом идёт целое число  $M$  ( $1 \leq M \leq 100\,000$ ) — количество туннелей. Далее вводятся  $M$  пар чисел — номера бункеров, соединенных туннелем. По каждому из туннелей можно двигаться в обе стороны. В организации не существует туннелей, ведущих из бункера в самого себя, зато может существовать более одного туннеля между парой бункеров.

### Формат выходных данных

В первой строке выведите  $N$  чисел, разделённых пробелом — для каждого из бункеров минимальное время, необходимое чтобы добраться до выхода. Считайте, что время перемещения по одному туннелю равно 1. Во второй строке выведите  $N$  чисел — для каждого бункера номер ближайшего бункера с выходом, если таких несколько выведите бункер с наименьшим номером.

### Примеры

<code>evacuation.in</code>	<code>evacuation.out</code>
3	1 0 1
1	2 2 2
2	
3	
1 2	
3 1	
2 3	

## Задача С. Цикл отрицательного веса

Имя входного файла: `negcycle.in`  
Имя выходного файла: `negcycle.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайта

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. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

### Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

## Задача D. После финала

Имя входного файла: `excursion.in`  
Имя выходного файла: `excursion.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Финал Чемпионата Бйгтландии по программированию впервые проводился в городе  $\mathcal{I}$ . Дорожная сеть города  $\mathcal{I}$  представляет собой  $N$  перекрёстков, соединённых  $M$  дорогами с двусторонним движением. Программа финала была столь насыщенной, что участники не успели осмотреть город. Более того, церемония закрытия затянулась, так что при отъезде команды-победителя времени оставалось только на то, чтобы добраться до аэропорта кратчайшим по суммарной длине путём. При этом, чтобы хотя бы немного посмотреть город, из всех таких маршрутов выбрали тот, который включает в себя целиком наибольшее количество дорог.

По заданной карте города вычислите длину маршрута и количество дорог, которое удалось посмотреть команде.

### Формат входных данных

Первая строка содержит два целых числа  $N$  и  $M$  — количество перекрёстков в городе  $\mathcal{I}$  и количество дорог соответственно ( $2 \leq N \leq 1000$ ,  $1 \leq M \leq 2 \cdot 10^5$ ).

Каждая из следующих  $M$  строк содержит три целых числа  $a_i$ ,  $b_i$  и  $c_i$  ( $1 \leq a_i, b_i \leq N$ ,  $a_i \neq b_i$ ,  $1 \leq c_i \leq 1000$ ) — номера перекрёстков, соединяемых  $i$ -й дорогой, и длину этой дороги. Два различных перекрёстка могут быть соединены более чем одной дорогой. Место проведения финала находится на перекрёстке с номером 1, аэропорт — на перекрёстке с номером  $N$ .

Гарантируется, что существует хотя бы один путь от места проведения финала в аэропорт.

### Формат выходных данных

Выведите два целых числа  $P$  и  $Q$  — длину кратчайшего пути от места проведения финала в аэропорт и максимальное число дорог, которые может включать в себя кратчайший путь.

### Примеры

<code>excursion.in</code>	<code>excursion.out</code>
3 3 1 2 1 1 3 2 2 3 1	2 2

## Задача Е. Правила дорожного движения

Имя входного файла: `forbidden.in`  
Имя выходного файла: `forbidden.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

В столице одной небольшой страны очень сложная ситуация. Многокилометровые пробки буквально парализовали движение в городе, и власти на многих улицах ввели одностороннее движение, не анализируя, можно ли будет теперь проехать из любого места в городе в любое другое, не нарушая правила. Транспортная система столицы представляет собой  $N$  площадей, соединенных  $M$  полосами для движения, в том числе круговыми полосами, проходящими по площади. Каждая полоса предназначена для движения только в одну определенную сторону. При этом на магистралях есть полосы, направленные как в одну, так и в другую сторону. По круговой полосе можно двигаться только внутри площади и только против часовой стрелки.

Власти города на каждой полосе разместили видеокамеру, поэтому если Иннокентий едет по встречной полосе (при ее наличии) или, в случае одностороннего движения, в сторону противоположную предписанной знаками, то после поездки против правил по каждой из полос ему придется заплатить штраф в размере одной тысячи тугриков этой страны.

Иннокентий, который торопится купить кафельную плитку со скидкой, решил доехать до магазина в любом случае, даже если для этого придется нарушать правила. Но он хочет выбрать такой маршрут движения, суммарный штраф на котором минимален.

Иннокентий еще не решил, откуда именно и в какой магазин он собирается ехать, поэтому ему необходимо ответить на несколько вопросов вида «Какой минимальный штраф надо заплатить, чтобы добраться из пункта  $A$  в пункт  $B$ ?». Отвечая на потребности жителей столицы, известная поисковая система Индекс разрабатывает соответствующий сервис.

Так как многие из вас рано или поздно будут проходить собеседование на работу в эту фирму, продемонстрируйте, что вы тоже умеете решать эту задачу.

### Формат входных данных

В первой строке входных данных содержатся два числа  $N$  и  $M$  — количество площадей и полос движения в городе соответственно ( $1 \leq N \leq 5000$ ,  $1 \leq M \leq 10\,000$ ). Далее содержатся описания полос, по которым движение разрешено. Каждая полоса описывается номерами двух площадей, которые она соединяет. Движение разрешено в направлении от первой из указанных площадей ко второй.

В следующей строке содержится одно число  $K$  — количество вопросов у Иннокентия ( $1 \leq K \leq 10\,000$ ,  $N \cdot K \leq 2 \cdot 10^7$ ). В следующих строках описываются вопросы, каждый вопрос описывается номерами двух площадей, между которыми требуется найти самый дешевый путь. Путь необходимо проложить от первой из указанных площадей ко второй.

### Формат выходных данных

Для каждого вопроса выведите одно число — искомый минимальный размер штрафа в тысячах тугриков. В случае, если пути между выбранной парой площадей не существует, выведите  $-1$ .

### Примеры

<code>forbidden.in</code>	<code>forbidden.out</code>
5 5	0
2 1	2
2 4	0
3 2	
4 3	
5 4	
3	
5 1	
1 5	
2 3	

## Задача F. Кратчайшие пути

Имя входного файла: path.in  
Имя выходного файла: path.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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 входных данных

Первая строка входного файла содержит три целых числа  $n$ ,  $m$ ,  $s$  — количество вершин и рёбер в графе и номер начальной вершины соответственно ( $2 \leq n \leq 2\,000$ ,  $1 \leq m \leq 5\,000$ ).

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 выходных данных

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

## Задача G. Егор и граф

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из  $n$  вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из  $n$  шагов.
- На  $i$ -том шаге Егор удаляет из графа вершину номер  $x_i$ . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как  $d(i, v, u)$  кратчайший путь между вершинами  $v$  и  $u$  в графе, который получился до удаления вершины  $x_i$ , то Егор хочет знать значение следующей суммы: 
$$\sum_{v, u, v \neq u} d(i, v, u).$$

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

### Формат входных данных

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 500$ ) — количество вершин в графе.

В следующих  $n$  строках содержится по  $n$  целых чисел — матрица смежности графа:  $j$ -тое число в  $i$ -той строке  $a_{ij}$  ( $1 \leq a_{ij} \leq 10^5, a_{ii} = 0$ ) обозначает вес ребра, ведущего из вершины  $i$  в вершину  $j$ .

В следующей строке содержится  $n$  различных целых чисел:  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq n$ ) — вершины, которые удаляет Егор.

### Формат выходных данных

Выведите  $n$  целых чисел —  $i$ -тое число равно искомой сумме перед  $i$ -тым шагом.

### Примеры

stdin	stdout
2 0 5 4 0 1 2	9 0

### Замечание

Задача взята с сайта <http://www.codeforces.com>

## Задача Н. Pink Floyd

Имя входного файла: floyd.in  
Имя выходного файла: floyd.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

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 входных данных

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 выходных данных

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	6
1 2 -2	5 6 5 7 2 3
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	