

## Задача А. Кратчайший путь между вершинами

Имя входного файла: `dist.in`  
Имя выходного файла: `dist.out`  
Ограничение по времени: 4 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 минимальный путь 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 ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 10\,000$ ).

### 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. Если путь есть, то вторая строка должна содержать одно целое неотрицательное число  $k$  — количество вершин в кратчайшем пути от  $s$  до  $t$ . В третьей строчке выведите  $k$  чисел — сам кратчайший путь. Если кратчайших путей несколько, выведите любой.

### Примеры

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

## Задача В. Цикл отрицательного веса

Имя входного файла: `negcycle.in`  
Имя выходного файла: `negcycle.out`  
Ограничение по времени: 3 секунды  
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 С. Заправки

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

В стране  $N$  городов, некоторые из которых соединены между собой дорогами. Для того, чтобы проехать по одной дороге, требуется один бак бензина. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в  $N$ -й, потратив как можно меньшее количество денег.

Дополнительно имеется канистра для бензина, куда входит столько же бензина, сколько входит в бак. В каждом городе можно заправить бак, залить бензин в канистру, залить и туда и туда, или же перелить бензин из канистры в бак.

### Формат входных данных

В первой строке дано число  $N$  ( $1 \leq N \leq 100$ ). Во второй строке входного файла записано  $N$  чисел,  $i$ -е из которых задает стоимость бензина в  $i$ -м городе (все числа целые в диапазоне от 0 до 100).

Во следующих строках описаны все дороги (по одной в строке). Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние, между двумя городами существует не более одной дороги, не существует дорог, ведущих из города в себя.

### Формат выходных данных

В выходной файл выведите одно число — суммарную стоимость маршрута или  $-1$ , если добраться до нужного города невозможно.

### Примеры

<code>gasstation.in</code>	<code>gasstation.out</code>
4 1 10 2 15 1 2 1 3 4 2 4 3	2

## Задача D. 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

<code>path.in</code>	<code>path.out</code>
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 Е. Подгорело

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Боря — обычный работник офиса с  $n$  кабинетами и  $m$  двухсторонними дорожками. Перемещаться между кабинетами в офисе можно только по дорожкам.

Сегодня необычный день, потому что в офисе начался пожар! Загорелся кабинет под номером  $f$ . А Борис в данный момент находится в кабинете под номером  $p$ . Также, в  $k$  кабинетах существуют пожарные выходы, добравшись до которых, Борис может спастись.

Пожар распространяется со скоростью один кабинет в минуту (т.е. через  $i$  минут будут гореть все кабинеты на расстоянии не более  $i$  от первоначальной точки возгорания). Боря распространяется с той же скоростью, но в отличие от пожара не заполняет собой все, что встречает на пути. Говоря человеческим языком, за одну минуту Боря может переместиться в кабинет, соединенный дорожкой с текущим.

При попадании в кабинет с выходом, Борис моментально спасается (разумеется, для удачного спасения нужно, чтобы кабинет еще не горел в это время). Если спустя  $t$  минут Борис и пожар оказываются в одном кабинете, то будем считать, что наш герой проигрывает в этой неравной схватке. Ваша задача определить, сможет ли Боря дойти до какого-либо выхода и эвакуироваться.

### Формат входных данных

На первой строке ввода даны два целых числа  $n$  ( $3 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ ) — количество кабинетов и дорожек соответственно.

В  $i$ -й из следующих  $m$  строк даны два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ , означающие, что  $i$ -я дорожка соединяет кабинеты с номерами  $u_i$  и  $v_i$ ).

На строке с номером  $m + 2$  дано единственное целое число  $f$  ( $1 \leq f \leq n$ ) — номер кабинета, в котором начался пожар. На строке  $m + 3$  дано единственное целое число  $p$  ( $1 \leq p \leq n$ ) — номер офиса Бориса.

На строке номер  $m + 4$  дано целое число  $k$  ( $0 \leq k \leq n - 2$ ) — количество выходов, после чего на следующих  $k$  строках дано по одному числу  $e_i$  ( $1 \leq e_i \leq n$ ) — номеру вершины  $i$ -го выхода.

Гарантируется, что выходы, место начала пожара и офис Бори являются попарно различными кабинетами.

### Формат выходных данных

Выведите «YES» (без кавычек), если Боря может спастись, и «NO» иначе.

## Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 1 2 4 1 3 1 4 1 3	NO
5 4 1 4 1 5 2 5 3 4 2 4 0	NO

## Задача F. 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  $i$ -м 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:  $j$ -е число в  $i$ -й 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 —  $i$ -е число равно искомой сумме перед  $i$ -м шагом.

### Примеры

stdin	stdout
2	9 0
0 5	
4 0	
1 2	

## Задача G. 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	

## Задача Н. Авиаперелёты-2

Имя входного файла: avia2.in  
Имя выходного файла: avia2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Профессору Форду необходимо попасть на международную конференцию. Он хочет потратить на дорогу наименьшее количество денег, поэтому решил, что будет путешествовать исключительно ночными авиарейсами (чтобы не тратиться на ночевку в отелях), а днем будет осматривать достопримечательности тех городов, через которые он будет проезжать транзитом. Он внимательно изучил расписание авиаперелетов и выбрал подходящие ему авиарейсы, выяснив, что перелеты на выбранных направлениях совершаются каждую ночь и за одну ночь он не сможет совершить два перелета.

Теперь профессор хочет найти путь наименьшей стоимости, учитывая что до конференции осталось  $K$  ночей (то есть профессор может совершить не более  $K$  перелетов).

### Формат входных данных

В первой строке находятся числа  $N$  (количество городов),  $M$  (количество авиарейсов),  $K$  (количество оставшихся ночей),  $S$  (номер города, в котором живет профессор),  $F$  (номер города, в котором проводится конференция). Ограничения:  $2 \leq N \leq 100$ ,  $1 \leq M \leq 10^5$ ,  $1 \leq K \leq 100$ ,  $1 \leq S \leq N$ ,  $1 \leq F \leq N$ .

Далее идет  $M$  строк, задающих расписание авиарейсов.  $i$ -я строка содержит три натуральных числа:  $S_i$ ,  $F_i$  и  $P_i$ , где  $S_i$  — номер города, из которого вылетает  $i$ -й рейс,  $F_i$  — номер города, в который прилетает  $i$ -й рейс,  $P_i$  — стоимость перелета  $i$ -м рейсом.  $1 \leq S_i \leq N$ ,  $1 \leq F_i \leq N$ ,  $1 \leq P_i \leq 10^6$ .

### Формат выходных данных

Выведите одно число — минимальную стоимость пути, подходящего для профессора. Если профессор не сможет за  $K$  ночей добраться до конференции, выведите число -1.

### Примеры

avia2.in	avia2.out
4 5 2 1 4 1 2 1 2 3 1 3 4 1 1 3 3 1 4 5	4

## Задача I. 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