

## Задача А. Хитрый министр снова с нами!

Входной файл: `minister.in`  
Выходной файл: `minister.out`  
Ограничение по времени: 2 секунды  
Ограничение памяти: 64 мегабайта

В Стране Дураков  $N$  городов, соединенных сетью из  $M$  автомобильных дорог. Движение по всем дорогам двустороннее и из любого города можно попасть в любой другой. Столица страны имеет номер 1. Каждая дорога в стране имеет некоторую целую неотрицательную длину.

В очередном своем ежегодном отчете Министерство Образования и Транспорта представило данные о длинах кратчайших путей из столицы страны до всех остальных городов. Поскольку эти данные готовились впопыхах, то большинство из них оказалось взято «с потолка». Для того, чтобы данные казались более правдоподобными, все объявленные расстояния были различными. К сожалению, Счетная Палата пожелала проверить деятельность Министерства и, в частности, истинность данных.

Поскольку данные о расстояниях уже опубликованы и стали достоянием общественности, то изменить их уже нет никакой возможности. Поэтому у министра есть единственный способ избежать отставки — в срочном порядке призвать дорожные службы и изменить длины дорог в стране так, чтобы заявленные расстояния соответствовали действительности.

Поскольку удлинение существующих дорог (а тем более уменьшение их длины!) занятие дорогостоящее, то министр вынужден искать такой план, при котором суммарное изменение всех длин было бы минимально. Конечно, никакие старания министра и дорожных служб не в силах сделать длину дороги отрицательной.

### Описание входного файла

В первой строке входного файла находятся числа  $N$  и  $M$  ( $1 \leq N \leq 4000$ ,  $1 \leq M \leq 100\,000$ ). Следующая строка содержит  $N$  различных целых чисел  $D_i$ , не превосходящих по модулю 100 000, — опубликованные расстояния от столицы до городов. Следующие  $M$  строк содержат описание существующих дорог — города, которые соединяет дорога, и ее длина (длина дороги задается неотрицательным целым числом, не превышающим 10 000).

### Описание выходного файла

Если изменить длины дорог указанным способом невозможно, выведите на первой строке выходного файла число  $-1$ . В противном случае выведите минимальное возможное суммарное изменение длины дорог. На следующих  $M$  строках выведите новые

длины дорог.

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

## Задача В. В поисках невест

Входной файл: `brides.in`  
Выходной файл: `brides.out`  
Ограничение по времени: 2 секунды  
Ограничение памяти: 64 мегабайта

Однажды король Флатландии решил отправить  $k$  своих сыновей на поиски невест. Всем известно, что во Флатландии  $n$  городов, некоторые из которых соединены дорогами. Король живет в столице, которая имеет номер 1, а город с номером  $n$  знаменит своими невестами.

Итак, король повелел, чтобы каждый из его сыновей добрался по дорогам из города 1 в город  $n$ . Поскольку, несмотря на обилие невест в городе  $n$ , красивых среди них не так много, сыновья опасаются друг друга. Поэтому они хотят добраться до цели таким образом, чтобы никакие два сына не проходили по одной и той же дороге (даже в разное время). Так как король любит своих сыновей, он хочет, чтобы среднее время сына в пути до города назначения было минимально.

### Описание входного файла

В первой строке входного файла находятся числа  $n$ ,  $m$  и  $k$  — количество городов и дорог во Флатландии и сыновей короля, соответственно ( $2 \leq n \leq 200$ ,  $1 \leq m \leq 2000$ ,  $1 \leq k \leq 100$ ). Следующие  $m$  строк содержат по три целых положительных числа каждая — города, которые соединяет соответствующая дорога и время, которое требуется

для ее прохождения (время не превышает  $10^6$ ). По дороге можно перемещаться в любом из двух направлений, два города могут быть соединены несколькими дорогами.

### Описание выходного файла

Если выполнить повеление короля невозможно, выведите на первой строке число  $-1$ . В противном случае выведите на первой строке минимальное возможное среднее время (с точностью 5 знаков после десятичной точки), которое требуется сыновьям, чтобы добраться до города назначения, не менее чем с пятью знаками после десятичной точки. В следующих  $k$  строках выведите пути сыновей, сначала число дорог в пути и затем номера дорог в пути в том порядке, в котором их следует проходить. Дороги нумеруются, начиная с единицы, в том порядке, в котором они заданы во входном файле.

brides.in	brides.out
5 8 2	3.00000
1 2 1	3 1 5 6
1 3 1	3 2 7 8
1 4 3	
2 5 5	
2 3 1	
3 5 1	
3 4 1	
5 4 1	

### Задача С. Обход в глубину

Входной файл: dfs.in  
Выходной файл: dfs.out  
Ограничение по времени: 1 секунды  
Ограничение памяти: 64 мегабайта

Всем вам хорошо известен алгоритм обхода в глубину. Будучи запущенный из некоторой вершины, он обходит все вершины графа, достижимые из нее, а заодно строит в графе ориентированное корневое дерево.

Конечно, какое именно из таких деревьев построит алгоритм зависит от того, в каком порядке он анализирует исходящие из каждой вершины дуги или, иными словами, в каком порядке для каждой вершины исходящие из нее дуги перечислены в списке смежности.

Вам задан ориентированный граф  $G$ , из вершины 1 которого достижимы все остальные вершины, и некоторое ориентированное корневое дерево  $T$  с корнем в вершине 1.

Необходимо выяснить, могло ли это дерево быть получено в результате обхода в глубину графа  $G$ , а если да, то в каком порядке в списках смежности должны идти для каждой вершины исходящие из нее дуги.

### Описание входного файла

В первой строке входного файла указано число вершин  $N$  в графе ( $1 \leq N \leq 10000$ ). Во второй строке записано  $N - 1$  число  $p_2 \dots p_N$ , где  $p_i$  обозначает предка вершины  $i$  в дереве  $T$  (вершина 1 служит корнем, поэтому предка не имеет).

В последующих  $N$  строках описаны списки смежности графа.  $i$ -я строка имеет вид:

$$k \ a_1 \ \dots \ a_k$$

Такая запись означает, что всего из вершины  $i$  исходят  $k$  дуг в вершины  $a_1, \dots, a_k$ . В графе могут быть петли и кратные дуги. Общее число дуг в графе не превышает 100 000.

Гарантируется, что число дуг в графе не превосходит 500 000.

### Описание выходного файла

В выходной файл в первую строку выведите сообщение YES, если дерево  $T$  может быть деревом обхода в глубину при подходящем переупорядочении дуг, и NO в противном случае. Если ответ положительный, то в последующих  $N$  строках необходимо описать, в каком порядке должны быть расположены исходящие дуги в списках смежности в том же формате, как они были заданы во входных данных.

### Пример

dfs.in	dfs.out
4	YES
1 4 1	2 4 2
2 2 4	1 3
1 3	0
0	1 3
1 3	
3	NO
1 1	
2 2 3	
1 3	
1 2	