

Задача А. Почтальон

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

В городе \mathfrak{C} есть n площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно прийти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, чтобы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество площадей в городе ($1 \leq n \leq 1000$). Далее следуют n строк, задающих улицы. В i -й из этих строк находится число m_i — количество улиц, выходящих из площади i . Далее следуют m_i пар положительных чисел. В j -й паре первое число — номер площади, в которую идет j -я улица с i -й площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят 10^5 , общее количество улиц также не превосходит 10^5 .

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте (считая первую и последнюю), а во вторую — номера площадей в порядке их посещения.

Если решений нет, выведите в выходной файл одно число -1 .

Если решений несколько, выведите любое.

Примеры

<code>post.in</code>	<code>post.out</code>
4	5
2 2 1 2 2	1 2 4 3 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

Задача В. Самое дешёвое ребро

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

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на M запросов вида «найти у двух вершин минимум среди стоимостей ребер пути между ними».

Формат входных данных

В первой строке файла записано одно число — n (количество вершин).

В i -й из следующих $n - 1$ строк записано два целых числа x и y ($x \leq i$, $|y| \leq 10^6$) — предок вершины i и стоимость ребра.

Далее следуют m ($0 \leq m \leq 5 \cdot 10^4$) запросов вида (x, y) — найти минимум на пути из x в y ($x \neq y$).

Формат выходных данных

Выведите m ответов на запросы.

Примеры

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

Задача С. LCA - 2

Имя входного файла: lca2.in
Имя выходного файла: lca2.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Задано подвешенное дерево, содержащее n вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z .

Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Корень дерева имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел это предок вершины i

Третья строка содержит целые числа a_1 и a_2 ($0 \leq a_i \leq n - 1$).

Четвёртая строка содержит три целых числа: x, y и z ($0 \leq x, y, z \leq 10^9$)

Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

lca2.in	lca2.out
3 2 0 1 2 1 1 1 0	2
1 2 0 0 1 1 1	0

Задача D. Учиться!

Имя входного файла: `moscow.in`
Имя выходного файла: `moscow.out`
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 64 мегабайта

Каждый год огромное количество выпускников, сдавшие ЕГЭ, выбирают, куда же они пойдут учиться. Не удивительно, что многие из них предпочитают перебраться поближе к столице. Транспортная инфраструктура страны переживает не лучшие времена, и в приемлемом качестве поддерживается минимально возможное число городов, необходимое для того, чтобы от любого города можно было добраться до любого другого.

Каждый выпускник оценивает свои результаты сдачи экзаменов, и решает, насколько далеко от своего родного города в сторону столицы он сможет уехать.

Выпускников настолько много, что вам не требуется выводить для каждого из них, до какого города он сможет доехать. Достаточно вывести сумму ответов для каждого выпускника.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$. В i -м запросе первое число соответствует городу, в котором окончил школу i -й выпускник, а второе — насколько далеко от родного города он может уехать. Все выпускники стараются перебраться как можно ближе к столице.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Столица имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру следующего за городом i на пути к столице. Третья строка содержит два целых числа в диапазоне от 0 до $n - 1$: a_1 и a_2 . Четвертая строка содержит три целых числа: x, y и z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров городов — ответов на все запросы.

Примеры

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

Задача E. Разбиение на пути

Имя входного файла: `paths.in`
Имя выходного файла: `paths.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество рёберно-простых путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leq n \leq 20\,000$) и число рёбер m ($1 \leq m \leq 20\,000$).
Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i и b_i ($1 \leq a_i, b_i \leq n$). Между каждыми двумя вершинами не более одного ребра. Граф связан.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей строке описание очередного пути – номера вершин в порядке прохождения.

Примеры

<code>paths.in</code>	<code>paths.out</code>
7 7	3
1 2	2 4 7 6
4 1	7 5
6 7	4 1 2 3
5 7	
7 4	
2 3	
4 2	

Задача F. Таня и пароль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из $n + 2$ символов. Она выписала все возможные n трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось n бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 2 \cdot 10^5$), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие n строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa