

Задача А. Папа Коли

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

Папа у Коли работает в фирме «Macrohard». С его слов Коля знает, что в этой всемирно известной компании работает N человек, причем Колин папа — главный начальник. Работники этой компании имеют привычку поручать задания своим подчинённым. При этом у каждого работника может быть произвольное число непосредственных подчинённых, но у каждого работника есть только один непосредственный начальник, который может отдавать ему поручения.

Недавно в компании решили провести реорганизацию, для чего каждому работнику необходимо узнать количество своих подчинённых (не обязательно непосредственных).

Формат входных данных

В первой строке входного файла находится число N — количество работников в фирме ($1 \leq N \leq 20\,000$). Вторая строка содержит $N - 1$ число от 1 до N : i -е число задаёт номер непосредственного начальника работника с номером $i + 1$. Безусловно, папа имеет в компании номер 1.

Формат выходных данных

Выходной файл должен содержать N чисел: i -е число должно быть равно количеству подчинённых i -го работника.

Примеры

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

Задача В. Дерево деревьев

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Однажды ночью, когда Артём и Андрей торопились в баню, они вспомнили одну историю. Шёл как-то Сергей Андреевич по просторам Берендеевых полян и увидел дерево, высокое-превысокое. Он посмотрел на него и задумался, а сколько других деревьев может быть на нём? Ребята посмеялись над этой историей, ведь на дереве не может быть других деревьев. Но им в голову пришла гениальная идея - сделать собственное уникальное дерево деревьев. Вы не поверите, они его сделали!!! Но теперь перед ними встал вопрос, а сколько же деревьев произрастает на их поделке? Итак, дано дерево. Нужно посчитать количество множеств вершин этого дерева, которые сами образуют дерево

Формат входных данных

В первой строке входного файла находится число n ($1 \leq n \leq 2 \cdot 10^5$) - количество вершин. В следующих $n - 1$ строках описываются веточки в их поделке. Каждая веточка задаётся парой чисел a, b ($1 \leq a \leq n, 1 \leq b \leq n, a \neq b$). Гарантируется, что у ребят получилось именно дерево

Формат выходных данных

Выходной файл должен содержать ответ на задачу - число различных непустых множеств вершин исходного дерева, которые сами образуют дерево. Так как ответ может быть достаточно большим, нужно вывести его по модулю $10^9 + 7$

Примеры

стандартный ввод	стандартный вывод
4 1 2 1 3 3 4	10
7 1 2 1 3 2 4 2 5 3 6 3 7	37

Задача С. Производство деталей

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

Предприятие «Авто-2010» выпускает двигатели для известных во всём мире автомобилей. Двигатель состоит ровно из n деталей, пронумерованных от 1 до n , при этом деталь с номером i изготавливается за p_i секунд. Специфика предприятия «Авто-2010» заключается в том, что там одновременно может изготавливаться лишь одна деталь двигателя. Для производства некоторых деталей необходимо иметь предварительно изготовленный набор других деталей.

Генеральный директор «Авто-2010» поставил перед предприятием амбициозную задачу — за наименьшее время изготовить деталь с номером 1, чтобы представить её на выставке.

Требуется написать программу, которая по заданным зависимостям порядка производства между деталями найдёт наименьшее время, за которое можно произвести деталь с номером 1.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 10^5$) — количество деталей двигателя.

Вторая строка содержит n натуральных чисел p_1, p_2, \dots, p_n , определяющих время изготовления каждой детали в секундах. Время для изготовления каждой детали не превосходит 10^9 секунд.

Каждая из последующих n строк входного файла описывает характеристики производства деталей. Здесь i -я строка содержит число деталей k_i , которые требуются для производства детали с номером i , а также их номера. В i -й строке нет повторяющихся номеров деталей. Сумма всех чисел k_i не превосходит $2 \cdot 10^5$.

Известно, что не существует циклических зависимостей в производстве деталей.

Формат выходных данных

В первой строке выходного файла должны содержаться два числа: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1 и число k деталей, которые необходимо для этого произвести.

Во второй строке требуется вывести через пробел k чисел — номера деталей в том порядке, в котором следует их производить для скорейшего производства детали с номером 1.

Примеры

стандартный ввод	стандартный вывод
3 100 200 300 1 2 0 2 2 1	300 2 2 1
2 2 3 1 2 0	5 2 2 1
4 2 3 4 5 2 3 2 1 3 0 2 1 3	9 3 3 2 1

Задача D. Робот в серверной

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

Вы — системный администратор одной очень перспективной IT-фирмы. В чём именно заключаются перспективы этой фирмы никто не знает, но все уверены, что они есть. Под вашим управлением находится серверная, в которой установлено целых n серверов (или n целых серверов, вы точно не уверены), пронумерованных целыми числами от 1 до n .

Ваша работа — следить за тем, чтобы серверы работали и не выключались. Если какой-нибудь сервер выключится, вы должны моментально его включить (по крайней мере, так считает ваш начальник). Вашу работу усложняет тот факт, что работоспособность некоторых серверов зависит от других серверов (например, для работы Web-сервера должен работать сервер баз данных). Если попытаться включить сервер, когда работают не все сервера, от которых он зависит, то этот сервер просто не включится (и потом придется пытаться включать его заново). Например, если сервер номер 1 зависит от серверов 2, 3 и 5, то для того, чтобы его включить, требуется сначала включить серверы 2, 3 и 5.

Как истинный системный администратор, вы ленивы и поэтому решили переложить эту сложную задачу на плечи маленького ни в чём не повинного робота. Бедняга робот не смог вам отказать, поэтому каждый день он выполняет следующий алгоритм. У робота в памяти записана последовательность из n различных чисел от 1 до n — порядок, в котором он обходит серверы. Подходя к очередному серверу, робот проверяет, выключен ли сервер, и если он выключен, то пытается его включить. После того, как робот дошёл до последнего сервера, он заканчивает выполнение алгоритма.

Недавно в серверной отключилось электричество, поэтому сейчас все серверы выключены. Вам интересно — сколько раз роботу придется выполнить свой алгоритм, чтобы все серверы оказались включенными.

Формат входных данных

Первая строка ввода содержит число n — количество серверов в серверной ($1 \leq n \leq 10^5$). Следующая строка содержит n различных чисел от 1 до n — порядок, в котором робот обходит серверы. В следующих n строках задано описание зависимостей между серверами: i -я строка содержит число k_i — количество серверов, от которых зависит i -й сервер, и последовательность k_i различных чисел — номера этих серверов. Сервер не может зависеть от самого себя. Суммарное количество зависимостей не превышает 1 000 000.

Формат выходных данных

В единственную строку выведите количество итераций алгоритма, через которое робот включит все серверы, или -1 , если робот так и не сможет включить все серверы.

Примеры

<code>servers.in</code>	<code>servers.out</code>
3 3 1 2 1 2 0 0	2
3 2 1 3 1 2 1 3 1 1	-1

Замечание

Подробное описание действий в первом примере:

1. Робот подходит к серверу 3 и включает его, так как этот сервер не зависит от других серверов;
2. Робот подходит к серверу 1 и не включает его, так как ещё не включен сервер 2, от которого зависит сервер 1;
3. Робот подходит к серверу 2 и включает его, так как этот сервер не зависит от других серверов;
4. Заканчивается первая итерация;
5. Робот подходит к серверу 3, сервер уже включен;
6. Робот подходит к серверу 1 и включает его, так как все серверы, от которых он зависит, уже включены;
7. Робот подходит к серверу 2, сервер уже включен;
8. Все серверы включены, прошло две итерации;

Во втором примере невозможно включить все серверы.

Задача Е. Два подграфа

Имя входного файла: two-subgraphs.in
Имя выходного файла: two-subgraphs.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Необходимо проверить, можно ли его вершины разбить на два множества так, чтобы подграфы, образуемые этими множествами, были полными.

Формат входных данных

В первой строчке входного файла записаны 2 числа N и M ($1 \leq n \leq 100$, $0 \leq m \leq \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что вершины i и j соединены ребром.

Формат выходных данных

В первой строке выходного файла выведите «YES», если возможно таким образом разбить вершины графа на 2 множества и «NO», если невозможно. Далее выведите сами множества в формате: в первой строке количество вершин в множестве, во второй— сами вершины в произвольном порядке.

Примеры

two-subgraphs.in	two-subgraphs.out
3 1	YES
1 3	2
	1 3
	1
	2

Путем в дереве называется последовательность вершин, в которой две соседние вершины соединены ребром. Все вершины в пути должны быть различны.

Для того, чтобы оценить красоту фейерверка, необходимо определить, какое максимальное количество вершин может содержать путь в дереве, которым представляется фейерверк. На рис. 3 приведен путь в дереве T^2 , содержащий максимальное количество вершин. Таким образом, красота фейерверка с базовым деревом T и мощностью 2 равна 10.

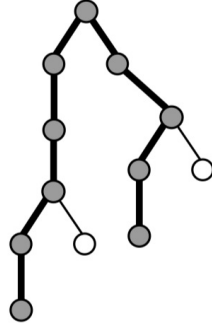


Рис. 3: Путь в дереве T^2 , содержащий максимальное количество вершин

Требуется написать программу, которая по описанию дерева T и натуральному числу m определяет красоту фейерверка с базовым деревом T и мощностью m .

Формат входных данных

В первой строке задано число n ($3 \leq n \leq 200\,000$) и число m ($1 \leq m \leq 200\,00$) — размер дерева и степень, в которую его требуется возвести соответственно.

Дальше следует строка, содержащая $n - 1$ число: p_2, \dots, p_n — предки соответствующих вершин дерева, $1 \leq p_i < i$

Корнем дерева является вершина с номером 1, гарантируется, что она не является листом.

Формат выходных данных

Выведите одно число — число вершин в диаметре дерева.

Примеры

стандартный ввод	стандартный вывод
4 2 1 1 2	10

Задача G. Космическая страна

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

Далекая-далекая космическая страна давным-давно состояла из N планет, причем столица страны находилась на планете с номером 1.

Так как не все жители страны могли позволить себе летать на звездолетах, на каждой планете находился вход ровно в один кибер-туннель, через который можно было попасть на некоторую другую планету. По кибер-туннелям можно перемещаться только в одну сторону (от входа к выходу). При этом туннели были созданы таким образом, что от каждой планеты всегда можно было добраться до столицы, используя некоторое количество туннелей.

Император этой страны часто проводил странные реформы, во время одной из них он решил изменить точки назначения для некоторых туннелей таким образом, чтобы от каждой планеты можно было добраться до столицы, совершив ровно K переходов по туннелям.

Для заданного описания туннелей страны, вам необходимо сказать, для какого минимального количества туннелей необходимо изменить точку выхода из туннеля, чтобы провести реформу.

Формат входных данных

В первой строке содержатся два числа N и K ($2 \leq N \leq 10^5$; $1 \leq K \leq 10^9$).

Во второй строке содержится N чисел a_i , каждое из которых задает точку выхода из туннеля со входом на планете i .

Формат выходных данных

Выведите единственное число — минимально возможное число туннелей, точку выхода которых нужно изменить, чтобы можно было добраться от любой планеты до столицы, используя ровно K туннелей.

Примеры

стандартный ввод	стандартный вывод
4 2 1 3 1 3	0
3 1 2 3 1	2