

Задача А. Компоненты связности - 2

Имя входного файла: `matrix2.in`
Имя выходного файла: `matrix2.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайта

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100\,000$), $0 \leq M \leq 100\,000$). 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 строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй — сами вершины в произвольном порядке.

Примеры

matrix2.in	matrix2.out
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

Задача В. 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 С. Построение

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

Группа солдат-новобранцев прибыла в армейскую часть №666. После знакомства с прапорщиком стало очевидно, что от работ на кухне по очистке картофеля спасти солдат может только чудо.

Прапорщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до N . После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прапорщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прапорщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прапорщик остался доволен.

Формат входных данных

Сначала на вход программы поступают числа N и M ($2 \leq N \leq 10^5$, $1 \leq M \leq 2 \cdot 10^5$) — количество солдат в роте и количество пар солдат, про которых прапорщик знает, кто из них выше.

Далее идут эти пары чисел A и B по одной на строке ($1 \leq A, B \leq N$), что означает, что, по мнению прапорщика, солдат A выше, чем B .

Не гарантируется, что все пары чисел во входных данных различны.

Формат выходных данных

В первой строке выведите «Yes» (если можно построиться так, чтобы прапорщик остался доволен) или «No» (если нет).

После ответа «Yes» на следующей строке выведите N чисел, разделенных пробелами, — одно из возможных построений.

Примеры

стандартный ввод	стандартный вывод
2 2 1 2 2 1	No
3 7 1 2 2 3 1 3 2 3 1 2 1 2 1 3	Yes 1 2 3

Задача D. Игрушечный лабиринт

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

Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, после чего останавливается. Целью игры является загнать шарик в одно из специальных отверстий-выходов. Шарик проваливается в отверстие, если оно встречается на его пути.

Первоначально шарик находится в левом верхнем углу лабиринта. Гарантируется, что решение существует и левый верхний угол не занят препятствием или отверстием.

Формат входных данных

В первой строке входного файла записаны числа N и M — размеры лабиринта (целые положительные числа, не превышающие 100). Затем идет N строк по M чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

Например, лабиринту, изображенному на рисунке, будет соответствовать такое описание:

	4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0
--	---

Формат выходных данных

Выведите единственное число — минимальное количество наклонов, которые необходимо сделать, чтобы шарик покинул лабиринт через одно из отверстий.

Примеры

labirint.in	labirint.out
4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0	3

Задача Е. Доставка кефирчика

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

Во время проведения очередной Межгалактической Летней Компьютерной Школы (МЛКШ) организаторы столкнулись с проблемой доставки кефирчика для вечерки. Дело в том, что кефирчик производят на планете под номером 1, а сами школьники живут на планете n , поэтому на доставку кефирчика тратится довольно большое время, а значит он успевает испортиться.

К счастью, галактическая транспортная система «Берендеев-Экспресс» постепенно внедряет новые кефиропроводы, способные передавать кефир со скоростью, в два раза превышающей скорость старых моделей. А именно, с любой планеты на любую по старым кефиропроводам кефир проходит за два года, а по новым — за один.

Разумеется, грешно было бы не воспользоваться инновационными технологиями, поэтому директор МЛКШ попросил вас написать программу, которая по данным о имеющихся кефиропроводах (как новых, так и старых) узнает кратчайший путь от планеты 1 до планеты n .

Формат входных данных

В первой строке входного файла даны два целых числа n и m ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$) — количество планет и количество кефиропроводов соответственно. В последующих m строках даны тройки натуральных чисел u_i , v_i и c_i . Числа u_i и v_i обозначают номера планет, соединенных i -м кефиропроводом, а c_i ($c_i = 1$ или $c_i = 2$) — количество лет, которое потребуется, чтобы передать кефир с одной планеты на другую через i -й кефиропровод. Планеты во входном файле нумеруются с единицы. Кефир по трубопроводам можно передавать в обоих направлениях.

Формат выходных данных

В выходной файл требуется вывести одно число — количество лет, которое требуется, чтобы доставить кефир с планеты 1 на планету n . Если доставка невозможна, то в выходной файл требуется вывести «-1».

Примеры

kefir.in	kefir.out
3 2 1 2 2 2 3 1	3
3 1 2 3 1	-1
2 5 1 2 1 1 2 2 1 2 1 1 1 2 2 2 1	1

Задача F. Числа

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

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

Примеры

<code>numbers.in</code>	<code>numbers.out</code>
9876	9876
8876	8769
	8768
	8876

Задача G. Красота фейерверка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В лаборатории теоретической пиротехники изучают новые технологии организации фейерверков. Фейерверк представляется как корневое дерево, а поскольку в мощном фейерверке его элементы также взрываются, порождая новые фейерверки, то ученые вводят операцию возведения корневого дерева в степень.

Корневое дерево содержит одну или несколько вершин. Одна из вершин выделена и называется корнем дерева, для каждой из остальных вершин ровно одна другая вершина является родителем. При этом от любой вершины можно Добраться До корня, последовательно переходя от вершины к ее родителю. Вершина, которая не является родителем никакой другой вершины, называется листом. Если вершина x является родителем вершины y , то вершина y является ребенком вершины x . Будем говорить, что вершина и ее родитель соединены ребром.

На рис. 1 показан пример корневого дерева с корнем в вершине 1. Родителем вершин 2 и 3 является вершина 1, родителем вершины 4 является вершина 2. Вершины 2 и 3 – дети вершины 1, а вершина 4 – ребенок вершины 2. Листьями являются вершины 3 и 4.

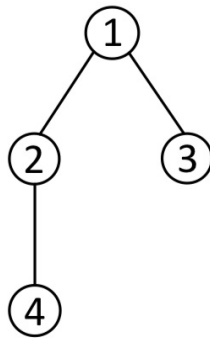


Рис. 1: Пример корневого дерева с корнем в вершине 1, листьями 3 и 4

Фейерверк задается своим базовым деревом T и мощностью m . Фейерверк представляется деревом, которое получается в результате возведения дерева T в степень m . Операция возведения дерева в степень устроена следующим образом. Если $m = 1$, то результат T^1 – само дерево T . Для $m > 1$ рассмотрим дерево T^{m-1} . Выполним следующую операцию: для каждого листа x дерева T^{m-1} создадим копию дерева T и назначим лист x родителем корня соответствующей копии. Получившееся дерево будет деревом T^m .

На рис. 2 показано дерево, представленное на рис. 1, в степенях 1, 2 и 3.

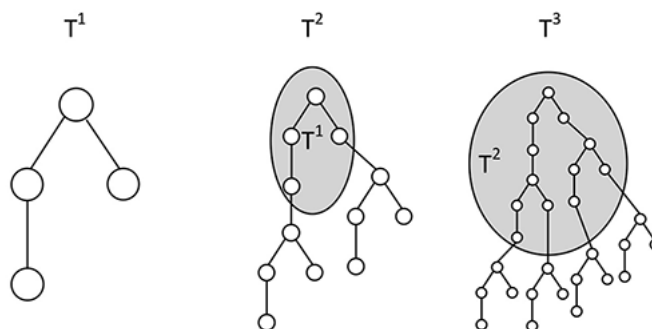


Рис. 2: Пример возведения дерева в степени 1, 2 и 3

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.

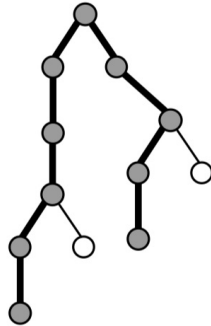


Рис. 3: Путь в дереве T^2 , содержащий максимальное количество вершин

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 Н. Только направо

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

Змей Горыныч оказался в лабиринте и хочет выбраться из него как можно скорее. К сожалению, после вчерашнего злоупотребления кефиром левая голова Змея соображает плохо. Поэтому Змей Горыныч никогда не поворачивается налево, а ещё ему после поворота направо нужно пройти минимум один шаг. Помогите Змею Горынычу определить длину кратчайшего пути до выхода из лабиринта.

Формат входных данных

В первой строке через пробел записаны числа r и c ($4 \leq r, c \leq 20$) — количество строк и столбцов в карте лабиринта. В каждой из следующих r строк записано по c символов, задающих эту карту. Символ `S` обозначает положение Змея Горыныча, символ `F` — точку выхода из лабиринта, символ `X` — стенку. Пробелами обозначены проходимые клетки. Гарантируется, что лабиринт окружен стенами. Перед началом движения Змей Горыныч может сориентироваться по любому из 4 направлений (вверх, вниз, влево или направо).

Формат выходных данных

Выведите единственное число — расстояние, которое придется пройти Змею Горынычу. Гарантируется, что он всегда сможет выйти из лабиринта.

Примеры

nolefts.in	nolefts.out
10 14 XXXXXXXXXXXXXXXX X XXX X XFXXXXX X XXX XX XX X X S X XX XXXXXX X X X X X X X X X X X XXX XX X XXXXXXXXXXXXXXXX	29

Замечание

Путь для теста из условия: $(5, 3) \rightarrow (5, 11) \rightarrow (9, 11) \rightarrow (9, 8) \rightarrow (7, 8) \rightarrow (7, 9) \rightarrow (8, 9) \rightarrow (8, 4) \rightarrow (3, 4)$