

Задача А. Компоненты связности - 2

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 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

stdin	stdout
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

Задача В. Долой списывание!

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 64 мегабайта

Во время теста Михаил Дмитриевич заметил, что некоторые лкшата обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день Михаил Дмитриевич был добрым, а потому решил разделить лкшат на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У Михаила Дмитриевича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

Формат входных данных

В первой строке находятся два числа N и M — количество лкшат и количество пар лкшат, обменивающихся записками ($1 \leq N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$). Далее в M строках расположены описания пар лкшат: два различных числа, соответствующие номерам лкшат, обменивающихся записками (нумерация лкшат идёт с 1). Каждая пара лкшат перечислена не более одного раза.

Формат выходных данных

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «YES»; иначе выведите «NO».

Примеры

stdin	stdout
3 2 1 2 2 3	YES

Задача С. Предок

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

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 100\,000$) — количество вершин в дереве. Во второй строке находятся n чисел, i -е из которых определяет номер непосредственного родителя вершины с номером i . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \leq m \leq 100\,000$) — количество запросов.

Каждая из следующих m строк содержит два различных числа a и b ($1 \leq a, b \leq n$).

Формат выходных данных

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a является одним из предков вершины b , и 0 в противном случае.

Примеры

ancestor.in	ancestor.out
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	

Задача D. Кратчайший путь

Имя входного файла: dag-shortpath.in
Имя выходного файла: dag-shortpath.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$, $1 \leq s, t \leq n$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите `Unreachable`.

Примеры

dag-shortpath.in	dag-shortpath.out
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable

Задача Е. Поиск цикла

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входных данных

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100\,000$, $M \leq 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Примеры

stdin	stdout
2 2 1 2 2 1	YES 2 1
2 2 1 2 1 2	NO

Задача F. Лягушачий пасьянс

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

Действие одной очень популярной компьютерной игры происходит на квадратном болоте размером N на N , разбитом на N^2 маленьких клеток-кувшинков.

Изначально на каждой кувшинке сидит лягушка одного из нескольких видов. Вид лягушки обозначается целым числом от 1 до 10^6 . Назовём стаей, содержащей данную лягушку, всех лягушек, до которых можно добраться от данной, перепрыгивая каждый раз на одну кувшинку по вертикали или горизонтали, не выходя за границы болота и прыгая только по лягушкам того же вида, что и данная.

При выборе некоторой лягушки автоматически выбираются все лягушки, входящие в одну стаю с ней. Если эта стая содержит хотя бы 2 лягушки, то все они дружно уплывают, и игроку начисляется $n \cdot (n - 1)$ очков, где n — количество лягушек в стае.

Задано начальное расположение лягушек на болоте. Необходимо для каждого вида лягушек, встречающегося на болоте, определить, какое максимальное количество очков можно набрать за первый ход, выбрав одну лягушку такого вида.

Формат входных данных

В первой строке входного файла `balls.in` задано целое число N — размер болота ($1 \leq N \leq 1000$).

В следующих N строках задано по N целых чисел, разделённых пробелами — виды лягушек на кувшинках. Каждое число — целое от 1 до 10^6 .

Формат выходных данных

Для каждого вида лягушек, встречающегося на болоте, выведите максимальное количество очков, которое можно набрать за первый ход, выбрав лягушку этого вида. Следуйте формату, приведённому в примере.

Примеры

balls.in	balls.out
11	1: 1190
1 1 1 1 1 2 2 2 3 3 3	2: 420
1 1 1 1 1 2 2 2 3 3 3	3: 420
1 1 1 1 1 2 2 2 3 3 3	4: 1056
1 1 1 1 1 2 2 2 3 3 3	5: 0
1 1 1 1 1 2 2 2 3 3 3	
1 1 1 1 1 2 2 2 3 3 3	
1 1 1 1 1 2 2 2 3 3 3	
4 4 4 4 4 4 4 4 4 4 4	
4 4 4 4 4 4 4 4 4 4 4	
4 4 4 4 4 4 4 4 4 4 4	
5 1 5 1 5 2 5 2 5 3 5	

Задача G. 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 Н. Два автобуса

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

Начинается летняя смена по информатике в ЛКШ. Все участники смены благополучно прилетели, и из аэропорта до центра их повезут на двух больших автобусах.

Организаторы смены решили заранее распределить участников по автобусам, чтобы ускорить процесс посадки. Но вскоре оказалось, что при распределении детей надо учитывать некоторые факторы. Из некоторых источников организаторы знают, что некоторые дети дружат, и они расстроятся, если посадить их в разные автобусы. И наоборот, некоторые дети недолюбливают друг друга, и сажать их в один автобус нельзя.

Учесть все эти ограничения достаточно сложно, поэтому организаторы попросили вас написать программу, которая получит корректное распределение участников по двум автобусам. Автобусы достаточно большие, чтобы вместить любое количество детей.

Формат входных данных

В первой строке заданы 3 числа n, m, k — число участников смены, количество пар участников которые должны оказаться в одном автобусе и количество пар участников, которые должны оказаться в разных автобусах ($1 \leq n \leq 10^5, 0 \leq m, k \leq 10^5$).

В следующих m строках перечислены пары номеров участников u_i и v_i , которые должны сидеть в одном автобусе ($1 \leq u_i, v_i \leq n, u_i \neq v_i$).

В следующих k строках перечислены пары номеров участников p_i и t_i , которые должны сидеть в разных автобусах ($1 \leq p_i, t_i \leq n, p_i \neq t_i$).

Формат выходных данных

Если требуемого распределения не существует, в единственной строке выведите «No» (без кавычек).

Иначе в первой строке выведите «Yes». Во второй строке выведите количество детей в одном автобусе, а затем их номера через пробел. В третьей строке выведите количество детей в другом автобусе, и их номера через пробел.

Примеры

стандартный ввод	стандартный вывод
3 1 1 1 2 2 3	Yes 2 1 2 1 3
3 0 3 1 2 2 3 1 3	No

Задача I. Наводнение

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

На базе «Берендеевы поляны» все домики соединены между собой дорожками. Каждая дорожка имеет свою высоту над уровнем озера Лель. Каждый домик может быть соединён с другим несколькими дорожками, но дорожек, ведущих из домика в него же, нет.

После сильных дождей озеро выходит из берегов и затапливает дорожки. Дорожка затапливается в том случае, если уровень воды больше или равен её высоте. Поскольку в лагере есть только самокаты и велосипеды, а лодок ещё не закупили, для нормального функционирования лагеря от любого домика до любого другого домика должен существовать путь по незатопленным дорожкам, возможно, по нескольким. Помогите сотрудникам базы определить минимальный уровень воды, при котором найдутся такие два домика, что от одного из них никак нельзя добраться до другого. Гарантируется, что до начала сильных дождей из любого домика можно было попасть в любой другой.

Формат входных данных

В первой строке даны два числа A и B — количество домиков и дорожек соответственно ($2 \leq A \leq 10\,000$, $1 \leq B \leq 20\,000$). Следующие B строк содержат тройки чисел (s_i, f_i, h_i) , где s_i и f_i — номера домиков, которые соединяет i -я дорожка, а h_i — её высота над уровнем озера в миллиметрах ($1 \leq s_i \leq A$, $1 \leq f_i \leq A$, $1 \leq h_i \leq 1\,000\,000$).

Формат выходных данных

Выведите минимальную высоту, на которую должна подняться вода, чтобы в лагере нашлось хотя бы два домика, между которыми нельзя пройти по незатопленным дорожкам.

Примеры

стандартный ввод	стандартный вывод
2 1 1 2 100	100
4 5 1 2 100 1 3 400 2 3 300 2 4 200 3 4 500	300

Задача J. Детали

Имя входного файла: `details.in`
Имя выходного файла: `details.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Предприятие «Авто-2005» 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-2005» 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-2005» 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 написано натуральное число n ($1 \leq n \leq 100\,000$). Вторая — n натуральных чисел $p_1, p_2, \dots, p_n$, определяющих время изготовления каждой детали в секундах ($1 \leq p_i \leq 1\,000\,000\,000$).

Каждая из последующих n строк входного файла описывает характеристики производства деталей. Здесь i -я строка содержит в начале количество деталей, которые требуются для производства детали с номером i , затем через пробел номера самих деталей. В списке нет повторяющихся номеров деталей. Сумма длин всех списков не превосходит 200 000.

Известно, что не существует циклических зависимостей в производстве деталей.

Формат выходных данных

В единственной строке выходного файла должно содержаться одно число: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1.

Примеры

details.in	details.out
3 100 200 300 1 2 0 2 2 1	300
4 2 3 4 5 2 3 2 1 3 0 2 1 3	9

Задача К. 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.

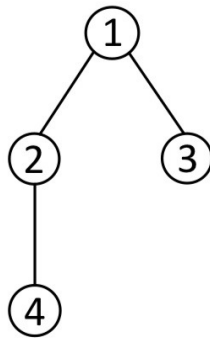


Рис. 1: Пример корневого дерева с корнем в вершине 1, листьями 3 и 4

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 назовем 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.

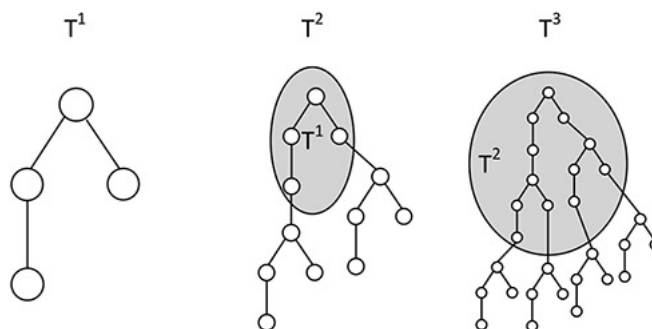


Рис. 2: Пример возведения дерева в степени 1, 2 и 3

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.

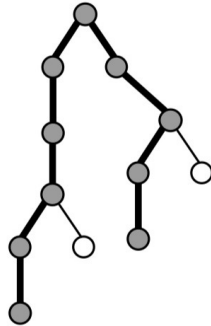


Рис. 3: Путь в дереве T^2 , содержащий максимальное количество вершин

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