

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. 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 С. Есть ли цикл?

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

Дан ориентированный граф. Требуется определить, есть ли в нем цикл.

Формат входных данных

Сначала вводятся два числа: N и M ($1 \leq N \leq 10^5, 0 \leq M \leq 10^5$) – количество вершин и ребер графа соответственно. Далее идет M пар чисел, задающих ребра.

Формат выходных данных

Выведите «-1» без кавычек, если нет цикла. В противном случае выведите в первую строку количество вершин в цикле, а во вторую — номера вершин в цикле в порядке обхода. Если ответов несколько выведите любой.

Примеры

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

Задача D. Два автобуса

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

Начинается осенняя смена по информатике в образовательном центре Сириус. 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 Е. 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 F. Противопожарная безопасность

Имя входного файла: `firesafe.in`
 Имя выходного файла: `firesafe.out`
 Ограничение по времени: 0.5 секунда
 Ограничение по памяти: 256 мегабайт

В Судиславле n домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Судиславле участились случаи пожаров. В связи с этим жители решили построить в посёлке несколько пожарных станций. Но возникла проблема: едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако возвращающаяся с задания машина обязана следовать правилам дорожного движения (жители Судиславля свято чтут эти правила!).

Ясно, что, где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой она выехала. Но строительство станций стоит больших денег, поэтому на совете посёлка было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 3000$). Во второй строке записано количество дорог m ($1 \leq m \leq 100\,000$). Далее следует описание дорог в формате $a_i b_i$, означающее, что по i -й дороге разрешается движение автотранспорта от дома a_i к дому b_i ($1 \leq a_i, b_i \leq n$).

Формат выходных данных

В первой строке выведите минимальное количество пожарных станций K , которое необходимо построить. Во второй строке выведите K чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

Примеры

firesafe.in	firesafe.out
5	2
7	4 5
1 2	
2 3	
3 1	
2 1	
2 3	
3 4	
2 5	

Задача G. Наводнение

Имя входного файла: `flood.in`
 Имя выходного файла: `flood.out`
 Ограничение по времени: 3 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

flood.in	flood.out
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 Н. 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.

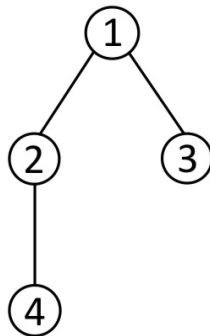


Рис. 1: Пример корневого дерева с корнем в вершине 1, листьями 3 и 4

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.

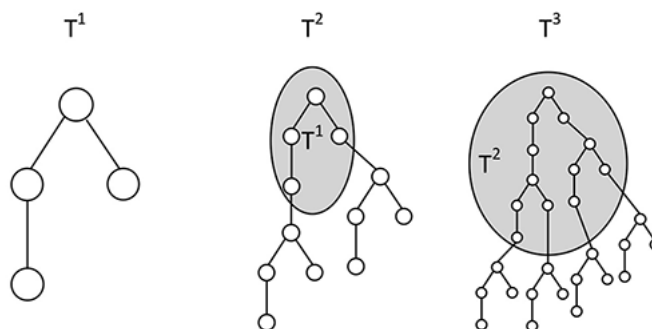


Рис. 2: Пример возведения дерева в степени 1, 2 и 3

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.

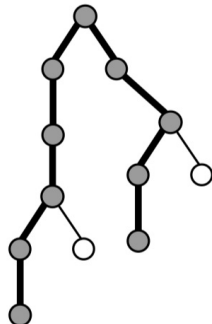


Рис. 3: Путь в дереве T^2 , содержащий максимальное количество вершин

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.

Система оценки

Первая группа: $3 \leq n \leq 5000$, $m = 1$

Вторая группа: $3 \leq n \leq 200\,000$, $m = 1$

Третья группа: $3 \leq n \leq 5000$, $1 \leq m \leq 5000$

Четвёртая группа: $3 \leq n \leq 5000$, $1 \leq m \leq 200\,000$

Пятая группа: $3 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$

Примеры

стандартный ввод	стандартный вывод
4 2 1 1 2	10

Задача I. Конденсация графа

Имя входного файла: `condense2.in`
Имя выходного файла: `condense2.out`
Ограничение по времени: 0.65 секунда
Ограничение по памяти: 256 мегабайт

Требуется найти количество рёбер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных рёбер и петель.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и рёбер графа соответственно ($n \leq 10\,000, m \leq 100\,000$). Следующие m строк содержат описание рёбер, по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — началом и концом ребра соответственно ($1 \leq b_i, e_i \leq n$). В графе могут присутствовать кратные рёбра и петли.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — количество рёбер в конденсации графа.

Примеры

condense2.in	condense2.out
4 4 2 1 3 2 2 3 4 3	2

Задача J. 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 К. 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

<code>dag-shortpath.in</code>	<code>dag-shortpath.out</code>
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable