

Задача А. Adjacency

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

Вам дан неориентированный граф. Требуется вывести его матрицу смежности.

Формат входных данных

В первой строке входного файла даны два натуральных числа n и m — количество вершин и количество рёбер графа ($1 \leq n \leq 100, 1 \leq m \leq 10000$). Вершины графа занумерованы числами от 1 до n . В следующих m строках даны пары чисел — концы очередного ребра. Каждое из этих чисел является натуральным и находится в пределах от 1 до n . Гарантируется, что между любыми двумя вершинами проходит не более одного ребра и в графе нет петель.

Формат выходных данных

Выведите в выходной файл n строк. В i -той строке должны оказаться n чисел. j -тое число i -той строки должно быть равно 1, если вершины i и j соединены ребро, и 0, если нет. На диагоналях матрицы должны стоять нули.

Примеры

стандартный ввод	стандартный вывод
1 0	0
2 1	0 1
1 2	1 0

Задача В. Поиск цикла

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

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

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3 1	YES 2 3 1
3 3 1 2 1 3 2 3	NO

Задача С. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$) — 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

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести -1.

Пример

стандартный ввод	стандартный вывод
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

Задача D. Гамильтонов путь

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

Дан ориентированный граф без циклов. Требуется проверить, существует ли в нем путь, проходящий по всем вершинам.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m — количество вершин и дуг графа соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — началом и концом дуги соответственно ($1 \leq b_i, e_i \leq n$).

Входной граф не содержит циклов и петель.

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

Формат выходных данных

Если граф удовлетворяет требуемому условию, то выведите YES, иначе NO.

Примеры

hamiltonian.in	hamiltonian.out
3 3 1 2 1 3 2 3	YES
3 2 1 2 1 3	NO

Задача Е. Точки сочленения

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

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 20\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i , e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество точек сочленения в заданном графе. На следующей строке выведите b целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

Пример

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

Задача F. Мосты

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

Дан неориентированный граф, не обязательно связный, но не содержащий петель и кратных рёбер. Требуется найти все мосты в нём.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 20\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

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

Задача G. Противопожарная безопасность

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

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

В последнее время в Марксе участились случаи пожаров. В связи с этим жители решили построить в городе 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 Маркса свято чтут эти правила!).

Ясно, что где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой выехала. Но строительство станций стоит больших денег, поэтому на совете города 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 ($1 \leq n \leq 3\,000$).

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 , которые необходимо построить. Во второй строке выведите K чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

Пример

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

Задача Н. Эвакуация

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

Одна из Сверхсекретных организаций, чье название мы не имеем право разглашать, представляет собой сеть из N подземных бункеров, соединенных равными по длине туннелями, по которым из любого бункера можно добраться до любого другого (не обязательно напрямую). Связь с внешним миром осуществляется через специальные засекреченные выходы, которые расположены в некоторых из бункеров. Организации понадобилось составить план эвакуации персонала на случай экстренной ситуации. Для этого для каждого из бункеров необходимо узнать, сколько времени потребуется для того, чтобы добраться до ближайшего из выходов. Вам, как специалисту по таким задачам, поручено рассчитать необходимое время для каждого из бункеров по заданному описанию помещения Сверхсекретной организации. Для вашего же удобства бункеры занумерованы числами от 1 до N .

Формат входных данных

В первой строке записано число N , во второй — число K ($1 \leq N \leq 100\,000$, $1 \leq K \leq N$) — количество бункеров и количество выходов соответственно. Далее через пробел записаны K различных чисел от 1 до N , обозначающих номера бункеров, в которых расположены выходы. Потом идёт целое число M ($1 \leq M \leq 100\,000$) — количество туннелей. Далее вводятся M пар чисел — номера бункеров, соединенных туннелем. По каждому из туннелей можно двигаться в обе стороны. В организации не существует туннелей, ведущих из бункера в самого себя, зато может существовать более одного туннеля между парой бункеров.

Формат выходных данных

В первой строке выведите N чисел, разделённых пробелом — для каждого из бункеров минимальное время, необходимое чтобы добраться до выхода. Считайте, что время перемещения по одному туннелю равно 1. Во второй строке выведите N чисел — для каждого бункера номер ближайшего бункера с выходом, если таких несколько выведите бункер с наименьшим номером.

Пример

evacuation.in	evacuation.out
3	1 0 1
1	2 2 2
2	
3	
1 2	
3 1	
2 3	

Задача I. Числа

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

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.

Пример

<code>numbers.in</code>	<code>numbers.out</code>
1234	1234
4321	2234
	3234
	4323
	4322
	4321

Задача J. Маяки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

В одном из государств было решено создать сеть из n городов, между которыми бы курсировали беспилотные автомобили, но остается проблема возникновения внештатных ситуаций. Для решения этого вопроса было решено поставить маяки, которые бы управляли автомобилями, в случаях возникновения опасности. Во всех n городах хотят поставить по одному маяку, и было решено, что, если какой-то один маяк выйдет из строя, то оставшиеся города тоже представляли из себя единую сеть, что из всех оставшихся существовал путь между любыми двумя городами. Так как государство не хочет сильно тратиться, то решили производить одинаковые маяки с наименьшим радиусом действия, но выполняющие все условия безопасности, которые были описаны ранее.

В нашем случае представим, что города — точки на плоскости, а дороги, как кратчайшее расстояние между точками. Заметим, что из города a в город b можно и через другие города. В итоге мы хотим выбрать минимальный радиус маяка, чтобы получить сеть, что из любого города можно добраться до любого другого и, если какой-то маяк выйдет из строя, то оставшиеся города тоже представляли из себя единую сеть.

Формат входных данных

Первая строка содержит одно целое число n ($0 \leq n \leq 300$) — количество городов. Следующие n строк содержат по два числа — координаты городов соответственно. Все координаты точек неотрицательные и не превосходят 10^9 .

Формат выходных данных

Выведите одно число: минимальный радиус действия для всех маяков, удовлетворяющий всем вышеописанным условиям.

Примеры

стандартный ввод	стандартный вывод
4 3 0 0 4 3 5 6 4	2.5000000000000000
6 2 1 3 5 8 2 8 4 12 1 12 5	3.041381265149110

Задача К. Бенни и Магический лабиринт

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

Бенни очутилась в Магическом лабиринте. Лабиринт состоит из комнат и дверей. Каждая дверь открывается только в одну сторону. В этом странном лабиринте бывают двери из комнаты в себя. Когда Бенни проходит через дверь Хозяин лабиринта дает ей монетку, причем монетки бывают разного номинала, но при открывании одной и той же двери Бенни получает всегда одинаковую монетку, иными словами номинал монетки — некоторая постоянная характеристика двери.

Бенни от скуки решила поиграть в следующую игру: она гуляет по лабиринту сколько угодно и хочет максимизировать разницу между самой дорогой и самой дешевой монетками.

Формат входных данных

Первая строка содержит два целых числа n и m — количество комнат и дверей в Магическом лабиринте, соответственно ($2 \leq n \leq 3 \cdot 10^5$; $1 \leq m \leq 3 \cdot 10^5$).

Следующий m строк содержат описания дверей — три целых числа v , u и w — описывают дверь, открывающуюся из комнаты v в комнату u с номиналом монетки w ($1 \leq v, u \leq n$; $0 \leq w \leq 10^9$).

Формат выходных данных

Выведите одно целое число — искомую разность.

Пример

стандартный ввод	стандартный вывод
6 8 1 2 9 2 3 7 3 1 1 2 4 5 3 4 4 4 5 6 5 6 3 6 4 10	9