

Задача А. Проверка на сильную связность

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Дан ориентированный граф. Проверить, является ли он сильно-связным.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 1000$) — количество вершин графа. Далее в n строках дана матрица смежности этого графа. В графе могут быть петли.

Формат выходных данных

Выведите YES, если граф сильно связан, и NO, если нет.

Пример

strongconcomp.in	strongconcomp.out
2 0 1 1 0	YES
4 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0	NO

Задача В. Компоненты сильной связности

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Дан связный ориентированный граф. Требуется найти компоненты сильной связности.

Формат входных данных

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

Следующие m строк содержат описание дуг по одной на строке. Дуга номер i описывается двумя натуральными числами b_i, e_i — номерами концов дуги ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число k — количество компонент сильной связности в заданном графе.

На следующей строке выведите n чисел — для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

Пример

condense.in	condense.out
6 7	2
1 2	1 1 1 2 2 2
2 3	
3 1	
4 5	
5 6	
6 4	
2 4	

Задача С. Конденсация графа

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Требуется найти количество ребер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных ребер.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leq 10\,000$, $m \leq 100\,000$). Следующие m строк содержат описание ребер, 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 ребра и петли.

Формат выходных данных

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

Пример

condense.in	condense.out
4 4 2 1 3 2 2 3 4 3	2

Задача D. Противопожарная безопасность

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

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

В последнее время в Зеленограде 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, возвращаясь с задания машина обязана следовать правилам дорожного движения (жители Зеленограда 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 ($1 \leq n \leq 3000$). Во второй строке записано количество дорог m ($1 \leq m \leq 100000$). 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 E. Авиаперелеты

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Главного конструктора Петю попросили разработать новую модель самолета для компании “Air Бубундия”. Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф “Air Бубундия” Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелета между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолет сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 1000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j -ое число в i -ой строке равно расходу топлива при перелете из i -ого города в j -ый. Все числа не меньше нуля и меньше 10^9 . Гарантируется, что для любого i в i -ой строчке i -ое число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Пример

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10