1

От списков смежности к матрице смежности

2

От матрицы смежности к спискам смежности

3

Граф с k компонентами связности

4

Минимальное и максимальное число ребер

5

Дорожная карта

6

Банкет

7

Компоненты связности - 2

8

Производство деталей

Задача 1.

Вводная

**От списков смежности к матрице смежности**

**Ограничение по времени работы программы: 1 секунда**

Постройте матрицу смежности по заданным спискам смежности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержится число N - количество вершин

(1⩽N⩽100). Далее идут N строк. В i-й строке содержится описание всех ребер, исходящих из i-й вершины. Описание начинается количеством исходящих ребер. Далее следуют номера вершин, в которые эти ребра идут. Все вершины нумеруются натуральными числами от 1 до N.

ВЫХОДНЫЕ ДАННЫЕ

Выведите N строк, содержащих по N чисел, равных 0 или 1, разделенных пробелами — матрицу смежности данного ориентированного графа.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 2 2 3 0 1 2 | 0 1 1 0 0 0 0 1 0 |

Начало формы

Задача решена верно

OK. Your score is = 100, 7/7 tests passed

Решение задачи

Будем считывать списки смежности по одному, строя матрицу смежности. Для каждого значения j из списка смежности i-й вершины необходимо сделать пометку A[i][j] = 1. Удобно вычитать из номеров вершины число 1 сразу же при построении матрицы смежности.

Конец формы

Задача 2.

Вводная

**От матрицы смежности к спискам смежности**

**Ограничение по времени работы программы: 1 секунда**

Простой ориентированный граф задан матрицей смежности. Выведите его представление в виде списков смежности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных находится число N — количество вершин графа (1⩽N⩽100). Далее идет N строк по N чисел в каждой, равных 0 или 1, содержащих матрицу смежности ориентированного графа из N вершин.

ВЫХОДНЫЕ ДАННЫЕ

Выведите N строк — списки смежности графа. В i-й строке сначала выведите количество исходящих из i-й вершины ребер, а затем — номера вершин, в которые эти ребра идут, упорядоченные по возрастанию.

Нумерация вершин начинается с числа 1.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 | 1 3 2 1 3 1 5 2 1 2 2 1 2 |

Начало формы

Задача решена верно

OK. Your score is = 100, 6/6 tests passed

Решение задачи

Сначала считайте матрицу смежности, потом постройте списки смежности по очереди для всех вершин. Список смежности для вершины i состоит из таких j, что A[i][j] == 1.

Конец формы

Задача 3.

Лёгкая

**Граф с k компонентами связности**

**Ограничение по времени работы программы: 1 секунда**

Постройте какой-нибудь неориентированный граф из N вершин содержащий K компонент связности.

ВХОДНЫЕ ДАННЫЕ

Даны два числа N и K, записанных в одной строке (1⩽K⩽N⩽100).

ВЫХОДНЫЕ ДАННЫЕ

Выведите матрицу смежности этого графа (она должна быть симметричной относительно главной диагонали, на главной диагонали должны стоять нули).

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 2 | 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 |

Задача 4.

Лёгкая

**Минимальное и максимальное число ребер**

**Ограничение по времени работы программы: 1 секунда**

Определите, какое минимальное и максимальное число ребер может быть в графе, содержащем N вершин и K компонент связности.

ВХОДНЫЕ ДАННЫЕ

Даны два числа N и K, записанных в одной строке (1⩽K⩽N⩽109).

ВЫХОДНЫЕ ДАННЫЕ

Выведите два числа — минимальное и максимальное возможное число ребер в таком графе.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 4 2 | 2 3 |

Начало формы

Загрузите исходный код программы

Python 3

Прикрепить файл

**Подсказка**

Минимальное число ребер если все компоненты связности - деревья.

**Подсказка**

Максимальное число ребер - если все компоненты связности - полные графы. При каком разделении N вершин на K полных подграфов общее число ребер будет максимальным?

**Подсказка**

Число ребер будет максимальным, если K−1 компонента связности будут содержать по одной вершине, а одна компонента связности будет полным подграфом из N−K+1 вершины.

Обратите внимание, что в программе на C++ для представления ответа необходимо использовать тип long long.

Конец формы

Задача 5.

Лёгкая

**Дорожная карта**

**Ограничение по времени работы программы: 2 секунды**

Дорожная карта у программиста Васи записана как матрица смежности графа. Напишите программу, которая поможет Васе определить, верно ли, что из первого города можно проехать во все остальные (не обязательно напрямую).

ВХОДНЫЕ ДАННЫЕ

В первой строке вводится количество городов на карте N ( 1 ≤ N ≤ 1000 ). В следующих N строках записано по N чисел, разделённых пробелами – элементы матрицы смежности графа, который описывает схему дорог.

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести слово 'YES', если из первого по порядку города можно проехать во все остальные, и 'NO', если это не так.

ПРИМЕРЫ

|  |  |
| --- | --- |
| **ВВОД** | **ВЫВОД** |
| 5 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 | YES |
| 5 0 1 1 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 | NO |

Задача 6.

Средняя

**Банкет**

**Ограничение по времени работы программы: 1 секунда**

На банкет были приглашены N Очень Важных Персон (ОВП). Были поставлены 2 стола. Столы достаточно большие, чтобы все посетители банкета могли сесть за любой из них. Проблема заключается в том, что некоторые ОВП не ладят друг с другом и не могут сидеть за одним столом. Вас попросили определить, возможно ли всех ОВП рассадить за двумя столами.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержатся два числа: N и M (1⩽N,M⩽100), где N — количество ОВП, а M — количество пар ОВП, которые не могут сидеть за одним столом. В следующих M строках записано по 2 числа — пары ОВП, которые не могут сидеть за одним столом.

ВЫХОДНЫЕ ДАННЫЕ

Если способ рассадить ОВП существует, то  выведите YES в первой строке и номера ОВП, которых необходимо посадить за первый стол, во второй строке. В противном случае в первой и единственной строке выведите NO.

ПРИМЕРЫ

|  |  |
| --- | --- |
| **ВВОД** | **ВЫВОД** |
| 3 2 1 2 1 3 | YES 1 |

Начало формы

Задача решена верно

OK. Your score is = 100, 10/10 tests passed

Решение задачи

Обозначим ОВП точками, а нелады между двумя ОВП — линиями. Получился граф. ОВП можно рассадить за двумя столами тогда и только тогда, когда этот граф двудольный. Граф двудольный тогда и только тогда, когда его вершины можно покрасить в два цвета так, чтобы ребра соединяли вершины, окрашенные в различные цвета. Осталось реализовать проверку на то, что вершины можно покрасить нужным образом. Сделаем это с помощью небольшой модификации обхода в глубину.

Конец формы

Задача 7.

Сложная

**Компоненты связности - 2**

**Ограничение по времени работы программы: 4 секунды**

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

ВХОДНЫЕ ДАННЫЕ

Первая строка входных данных содержит два целых числа N и M (0<N≤100000), 0≤M≤100000). В следующих M строках записаны по два числа i и j (1⩽i,j⩽N), которые означают, что вершины i и j соединены ребром.

ВЫХОДНЫЕ ДАННЫЕ

В первой строке выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в отдельной строке количество вершин в компоненте, в следующей строке — сами вершины через пробел в произвольном порядке.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 6 4 3 1 1 2 5 4 2 3 | 3 3 1 2 3 2 4 5 1 6 |

Задача 8.

Олимпиадная

**Производство деталей**

**Ограничение по времени работы программы: 2 секунды**

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 pi 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.

ВХОДНЫЕ ДАННЫЕ

Первая строка входных данных содержит число n (1⩽n⩽100000) — количество деталей двигателя. Вторая строка содержит n натуральных чисел p1, p2, …, pn, определяющих время изготовления каждой детали в секундах. Время для изготовления каждой детали не превосходит 109 секунд.

Каждая из последующих n строк входных данных описывает характеристики производства деталей. Здесь i-я строка содержит число деталей ki, которые требуются для производства детали с номером i, а также их номера. В i-й строке нет повторяющихся номеров деталей. Сумма всех чисел ki не превосходит 200000.

Известно, что не существует циклических зависимостей в производстве деталей.

ВЫХОДНЫЕ ДАННЫЕ

В первой строке программа должна вывести 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.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **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 |