**Отчет**

Выполнил студент Шакирзянов Руслан 4214

**Задача**

|  |  |  |
| --- | --- | --- |
| Алгоритм Флери | Одна из форм представления графов (матрица смежности, матрица инциденций, списки смежности) – по выбору | Эйлеровый цикл/путь |

**Процедурное решение**

1) задать эйлеров граф G;

2) положить текущий граф равным G, а текущую вершину -- равной произвольной вершине v;

3) выбрать произвольное, с учетом ограничения (см. ниже) ребро e текущего графа, инцидентное текущей вершине;

4) назначить текущей вторую вершину, инцидентную e;

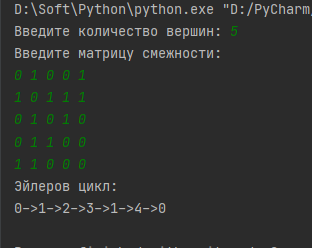
5) удалить e из текущего графа и внести в список;

6) если в текущем графе еще остались ребра, вернуться на шаг 2;

7) в результате получим список ребер графа G в той последовательности, в которой они образуют эйлеров цикл.

Ограничение: если степень текущей вершины в текущем графе больше 1, нельзя выбирать ребро, удаление которого из текущего графа увеличит число компонент связности в нем.

**Пример работы программы**



Код программы

def SmejToSmejList(vertex\_num, smejnost\_matrix):  
 smejnost\_lists = [[i] for i in range(vertex\_num)] # создание списка смежности с номерами вершин  
 for i in range(len(smejnost\_matrix)):  
 for j in range(len(smejnost\_matrix[i])): # проход по матрице смежности  
 if smejnost\_matrix[i][j] != 0: # если найдена смежная вершина  
 smejnost\_lists[i].append([j, smejnost\_matrix[i][j] if smejnost\_matrix[i][j] != 2 else 1]) # добавить в список смежности пару в формате [вершина, вес]  
 return smejnost\_lists  
  
  
def ToEdgeList(smejnost\_list): # функция создания списка ребер из списков смежности в элементами в формате [ребро1, ребро2, вес]  
 edge\_list = list()  
 for i in range(len(smejnost\_list)):  
 for j in range(len(smejnost\_list[i])):  
 if (i, smejnost\_list[i][j]) not in edge\_list and (smejnost\_list[i][j], i) not in edge\_list:  
 edge\_list.append((i, smejnost\_list[i][j]))  
 return edge\_list  
  
  
def Flery(smejnost\_list, in\_i=-1, out\_i=-1):  
 edge\_list = ToEdgeList(smejnost\_list) # список ребер  
 cur\_i = 0 # текущая вершина  
 out = [] # результирующий список  
 for i in range(len(smejnost\_list)):  
 if len(smejnost\_list[i]) != 0:  
 cur\_i = i # выбор такой текущей вершины, чтобы она была смежна с хотябы еще одной вериной  
 break  
 if in\_i != -1:  
 cur\_i = in\_i # если задана начальная вершина делаем ее текущей  
 while len(edge\_list) != 0: # пока в списке ребер есть ребра выполняем алгоритм  
 for cur\_o in smejnost\_list[cur\_i]: # проходим по всем смежным к текущей вершинам  
 temp\_copy = []  
 for i in range(len(smejnost\_list)):  
 temp = []  
 for j in smejnost\_list[i]: # создаем и заполняем копию списка смежности с удаленным ребром (cur\_i,cur\_o)  
 if not (i == cur\_i and j == cur\_o) and not (j == cur\_i and i == cur\_o):  
 temp.append(j)  
 temp\_copy.append(temp)  
 out.append(cur\_i)  
 if (cur\_i, cur\_o) in edge\_list:  
 edge\_list.remove((cur\_i, cur\_o)) # удаляем ребро из списка ребер  
 else:  
 edge\_list.remove((cur\_o, cur\_i))  
 smejnost\_list[cur\_i].remove(cur\_o) # удаляем вершины ребро из списка смежности  
 smejnost\_list[cur\_o].remove(cur\_i)  
 cur\_i = cur\_o # меняем текщее ребро  
 break  
 if in\_i == -1:  
 out.append(out[0])  
 print("Эйлеров цикл:\n" + '->'.join(str(i) for i in out))  
 else:  
 out.append(out\_i)  
 print("Эйлеров путь:\n" + '->'.join(str(i) for i in out)) # красивы вывод  
  
  
def main():  
 vertex\_num = int(input("Введите количество вершин: ")) # запрос навведенение количества вершин  
 print("Введите матрицу смежности:")  
 smejnost\_matrix = [[int(i) for i in input().split()] for j in  
 range(vertex\_num)] # ввод матрицы смежности с клавиатуры  
 smejnost\_list\_temp = SmejToSmejList(vertex\_num, smejnost\_matrix)  
 smejnost\_list = []  
 for i in range(vertex\_num):  
 temp\_list = []  
 for j in range(1, len(smejnost\_list\_temp[i])):  
 temp\_list.append(smejnost\_list\_temp[i][j][0]) # изменение структуры списка смежности из предыдущих работ  
 smejnost\_list.append(temp\_list)  
 if all(map(lambda x: len(x) % 2 == 0, smejnost\_list)): # проверка на эйлеров граф  
 Flery(smejnost\_list)  
 elif list(map(lambda x: len(x) % 2 == 0, smejnost\_list)).count(False) == 2: # проверка на полуэйлеров граф  
 in\_i = list(map(lambda x: len(x) % 2 == 0, smejnost\_list)).index(False) # задание начальной вершины с нечетным количеством смежных вершин  
 out\_i = list(map(lambda x: len(x) % 2 == 0, smejnost\_list)).index(False, in\_i + 1) # задание конечной такойже вершины  
 Flery(smejnost\_list, in\_i=in\_i, out\_i=out\_i)  
 else:  
 print("Не эйлеров граф")  
  
  
main()  
""" пример цикл  
5  
0 1 0 0 1  
1 0 1 1 1  
0 1 0 1 0  
0 1 1 0 0  
1 1 0 0 0  
  
примеры путь  
5  
0 1 0 0 1  
1 0 1 1 1  
0 1 0 1 0  
0 1 1 0 1  
1 1 0 1 0  
  
5  
0 1 0 0 1  
1 0 1 0 1  
0 1 0 1 0  
0 0 1 0 0  
1 1 0 0 0  
"""