**Отчет**

Выполнил студент Шакирзянов Руслан 4214

|  |  |  |
| --- | --- | --- |
| Алгоритм обхода в глубину  Алгоритм обхода в ширину | Одна из форм представления графов (матрица смежности, матрица инциденций, списки смежности) – по выбору | Для одного и того же графа показать обход в глубину и ширину |

**Задача**

**Процедурное решение**

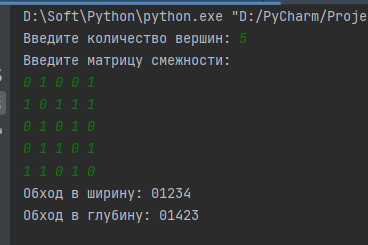
Обход в глубину

Реализуется при помощи стека. Устанавливаем текущую вершину, и проходимся по смежным с ней меняя текущую вершину на каждом шаге и также записываем их в стек, если дойдем до тупика возвращаемся по стеку до последней развилки и продолжаем до тех пор, пока по стек не опустеет, при этом записываем пройденные впервые вершины этот список и будет результатом обхода в глубину.

Обход в ширину

Реализуется при помощи очереди. В цикле проходим по всем вершинам и записываем смежные с ними в очередь если они там еще не присутствуют. На каждом шаге берем первую в очереди вершину и записываем ее в результирующий список. Когда очередь опустеет алгоритм завершается, а результирующий список будет результатом обхода в ширину.

**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 VGubinu(smejnost\_list): # функция прохода в глубину  
 out = [] # результирующий список  
 stek = [] # стек для реализации прохода в глубну  
 cur = 0 # номер текущей вершины  
 for i in range(len(smejnost\_list)):  
 if len(smejnost\_list[i]) != 0: # выбор такой текущей вершины, чтобы она была смежна с хотябы еще одной вериной  
 cur = i  
 break  
 out.append(cur) # ввод начальной вершины в результирующий список и стек  
 stek.append(cur)  
 out\_fl = True # флаг выхода из цикла  
 while out\_fl:  
 for i in smejnost\_list[cur]:  
 if i not in out:  
 cur = i # выбор новой вершины еще не записанной в результирующий список  
 break  
 if cur in out: # если на предыдущем шаге не найдена не использованная вершина вернутья на предыдущую вершину через стек  
 if len(stek) == 0: # если при этом стек пуст завершить цикл  
 out\_fl = False  
 continue  
 cur = stek.pop()  
 else: # иначе добавить новую вершину в результирующий список и стек  
 out.append(cur)  
 stek.append(cur)  
 print("".join(str(i) for i in out)) # вывод результата прохода в глубину  
  
  
def VShirinu(smejnost\_list): # функция прохода в ширину  
 out = [] # результирующий список  
 queue = [] # очередь для реализации прохода в ширину  
 cur = 0 # номер текущей вершины  
 for i in range(len(smejnost\_list)):  
 if len(smejnost\_list[i]) != 0: # выбор такой текущей вершины, чтобы она была смежна с хотябы еще одной вериной  
 cur = i  
 break  
 queue.append(cur) # добавление вершины в очередь  
 while len(queue) != 0:  
 cur = queue.pop(0) # получение вершины из начала очереди  
 for i in smejnost\_list[cur]: # проход по смежным с текущей вершинам  
 if i not in queue and i not in out: # если вершины нет в результирующем списке и в очереди то добавить ее в очередь  
 queue.append(i)  
 out.append(cur) # добавлен е текущей вершины в результирующий список  
 print("".join(str(i) for i in out)) # вывод результата прохода в ширину  
  
  
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)  
 print("Обход в ширину: ", end='')  
 VGubinu(smejnost\_list)  
 print("Обход в глубину: ", end='')  
 VShirinu(smejnost\_list)  
  
  
main()  
""" пример  
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  
"""