**Отчет**

Выполнил студент Шакирзянов Руслан 4214

**Задача**

|  |  |  |
| --- | --- | --- |
| Алгоритм Дейкстры | Одна из форм представления графов (матрица смежности, матрица инциденций, списки смежности) – по выбору | Кратчайший путь от одной вершины до всех остальных вершин графа |

**Процедурное решение**

Каждой вершине из V сопоставим метку — минимальное известное расстояние от этой вершины до a.

Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки.

Работа алгоритма завершается, когда все вершины посещены.

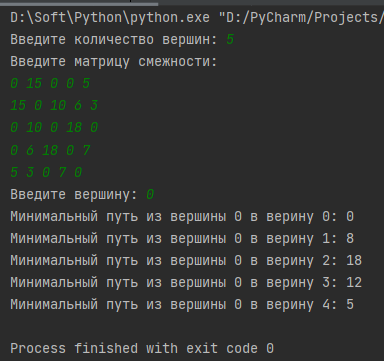
Инициализация.

* Метка самой вершины a полагается равной 0, метки остальных вершин — бесконечности.
* Это отражает то, что расстояния от a до других вершин пока неизвестны.
* Все вершины графа помечаются как непосещённые.

Шаг алгоритма.

* Если все вершины посещены, алгоритм завершается.
* В противном случае, из ещё не посещённых вершин выбирается вершина u, имеющая минимальную метку.
* Мы рассматриваем всевозможные маршруты, в которых u является предпоследним пунктом. Вершины, в которые ведут рёбра из u, назовём соседями этой вершины. Для каждого соседа вершины u, кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки u и длины ребра, соединяющего u с этим соседом.
* Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину u как посещённую и повторим шаг алгоритма.

**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 словаря ребер из списков смежности с с элементами по индексам (ребро1б ребро2) и значениями весов  
 edge\_list = dict()  
 for i in range(len(smejnost\_list)):  
 for j in smejnost\_list[i]:  
 if (i, j[0]) not in edge\_list and (j[0], i) not in edge\_list:  
 edge\_list[(i, j[0])] = j[1]  
 return edge\_list  
  
  
def ToNevzvesh(smejnost\_list): # функция перевода взвешенных списков смежности в невзвешанные  
 out = []  
 for i in smejnost\_list:  
 temp\_list = []  
 for j in i:  
 temp\_list.append(j[0]) # преобразование пары (вершина, вес) в просто элемент вершина  
 out.append(temp\_list)  
 return out  
  
  
def Deykstra(smejnost\_list\_v, vertex):  
 deykstra = [-1 for \_ in range(len(smejnost\_list\_v))] # заполнение выходного списка длин -1  
 deykstra[vertex] = 0 # длинна заданной вершины 0  
 end\_flag = True # флаг выхода из цикла  
 cur = vertex # текущая вершина  
 fixed = [] # список зафиксированных длинн вершин  
 while end\_flag:  
 for i in smejnost\_list\_v[cur]: # проход по всем смежным с текущей вершинам и заполнение списка длин добавлением весов  
 deykstra[i[0]] = i[1] + deykstra[cur] if (i[1] + deykstra[cur]) < deykstra[i[0]] \  
 or deykstra[i[0]] == -1 else deykstra[i[0]]  
 fixed.append(cur) # фиксируем текущаю вершину  
 cur = min(deykstra[i[0]] if i[0] not in fixed else 100\*\*100 for i in smejnost\_list\_v[cur]) # проверка на возожности продолжение выполнения алгоритма  
 if cur == 100\*\*100:  
 end\_flag = False  
 else:  
 cur = deykstra.index(cur) # замена текущей вершины  
 for i in range(len(deykstra)):  
 print(f"Минимальный путь из вершины {vertex} в верину {i}: {deykstra[i]}") # красивый 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\_v = []  
 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]) # изменение структуры списка смежности из предыдущих работ  
 smejnost\_list\_v.append(temp\_list)  
 vertex = int(input("Введите вершину: ")) # ввод начальной вершины  
 Deykstra(smejnost\_list\_v, vertex)  
  
  
main()  
""" пример  
5  
0 15 0 0 5  
15 0 10 6 3  
0 10 0 18 0  
0 6 18 0 7  
5 3 0 7 0  
0  
"""