

Алгоритм Дейкстры

Этот алгоритм позволяет найти кратчайшие пути из вершины графа до всех доступных. Также с помощью него можно определить, является ли граф связным.

Данная реализация предназначена для работы на неориентированных графах без петель и кратных ребер.

Основные шаги

- 1) Читаем матрицу весов, индекс стартовой вершины
- 2) Выделяем вершину как стартовую (то есть путь до нее состоит только из нее самой, и длина этого пути равна нулю)
- 3) Задаем веса (равные весам ребер из стартовой вершины) всем доступным из стартовой *[и неисследованным]* вершинам
- 4) Отмечаем стартовую как исследованную
- 5) Ищем среди уже посещенных *[но еще не исследованных]* вершину с наименьшим весом (либо же любую из вершин с наименьшим весом), делаем ее новой стартовой
- 6) Повторяем пункты 3)-5) пока, на 5) находится хотя бы подходящая вершина

Формат input:

размер матрицы *номер стартовой вершины (нумерация с нуля)*

матрица

input	output
Первая матрица (связная)	
5 0 0 1 3 5 1 1 0 0 4 0 3 0 0 2 1 5 4 2 0 3 1 0 1 3 0	These nodes are reachable from the starting node: #0 way size = 0 way = [0] #1 way size = 1 way = [0, 1] #2 way size = 2 way = [0, 4, 2] #3 way size = 4 way = [0, 4, 3] #4 way size = 1 way = [0, 4]
5 3 0 1 3 5 1 1 0 0 4 0 3 0 0 2 1 5 4 2 0 3 1 0 1 3 0	These nodes are reachable from the starting node: #0 way size = 4 way = [3, 4, 0] #1 way size = 4 way = [3, 1] #2 way size = 2 way = [3, 2] #3 way size = 0 way = [3] #4 way size = 3 way = [3, 4]
Вторая матрица (полный граф)	
5 0 1	These nodes are reachable from the starting node: #0 way size = 0 way = [0] #1 way size = 1 way = [0, 1] #2 way size = 1 way = [0, 2] #3 way size = 1 way = [0, 3] #4 way size = 1 way = [0, 4]

Третья матрица (из трех компонент)

11 0
0 1 3 5 1 0 0 0 0 0 0
1 0 0 4 0 0 0 0 0 0 0
3 0 0 2 1 0 0 0 0 0 0
5 4 2 0 3 0 0 0 0 0 0
1 0 1 3 0 0 0 0 0 0 0
0 0 0 0 0 0 2 8 2 0 0
0 0 0 0 0 2 0 3 0 5 0
0 0 0 0 0 8 3 0 1 0 0
0 0 0 0 0 2 0 1 0 7 0
0 0 0 0 0 0 5 0 7 0 0
0 0 0 0 0 0 0 0 0 0 0

These nodes are reachable from the starting node:
#0 way size = 0 way = [0]
#1 way size = 1 way = [0, 1]
#2 way size = 2 way = [0, 4, 2]
#3 way size = 4 way = [0, 4, 3]
#4 way size = 1 way = [0, 4]
These nodes are in another graph component(s):
[5, 6, 7, 8, 9, 10]

11 6
0 1 3 5 1 0 0 0 0 0 0
1 0 0 4 0 0 0 0 0 0 0
3 0 0 2 1 0 0 0 0 0 0
5 4 2 0 3 0 0 0 0 0 0
1 0 1 3 0 0 0 0 0 0 0
0 0 0 0 0 0 2 8 2 0 0
0 0 0 0 0 2 0 3 0 5 0
0 0 0 0 0 8 3 0 1 0 0
0 0 0 0 0 2 0 1 0 7 0
0 0 0 0 0 0 5 0 7 0 0
0 0 0 0 0 0 0 0 0 0 0

These nodes are reachable from the starting node:
#5 way size = 2 way = [6, 5]
#6 way size = 0 way = [6]
#7 way size = 3 way = [6, 7]
#8 way size = 4 way = [6, 5, 8]
#9 way size = 5 way = [6, 9]
These nodes are in another graph component(s):
[0, 1, 2, 3, 4, 10]

11 10
0 1 3 5 1 0 0 0 0 0 0
1 0 0 4 0 0 0 0 0 0 0
3 0 0 2 1 0 0 0 0 0 0
5 4 2 0 3 0 0 0 0 0 0
1 0 1 3 0 0 0 0 0 0 0
0 0 0 0 0 0 2 8 2 0 0
0 0 0 0 0 2 0 3 0 5 0
0 0 0 0 0 8 3 0 1 0 0
0 0 0 0 0 2 0 1 0 7 0
0 0 0 0 0 0 5 0 7 0 0
0 0 0 0 0 0 0 0 0 0 0

These nodes are reachable from the starting node:
#10 way size = 0 way = [10]
These nodes are in another graph component(s):
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Четвертая матрица (из 15 компонент)

15 0

000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

These nodes are reachable from the starting node:
#0 way size = 0 way = [0]
These nodes are in another graph component(s):
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]