Практическая работа № 2

студента группы ИТз-221

Дмитриев Дмитрий Анатольевич

Выполнение:\_\_\_\_\_\_\_\_\_ Защита:\_\_\_\_\_\_\_\_\_\_\_

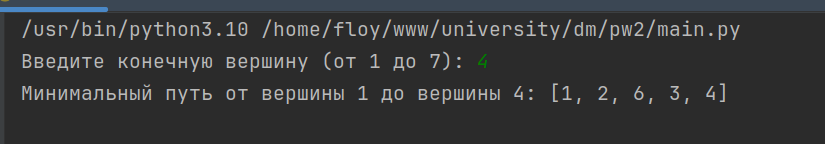
**Алгоритм Дейкстры.**

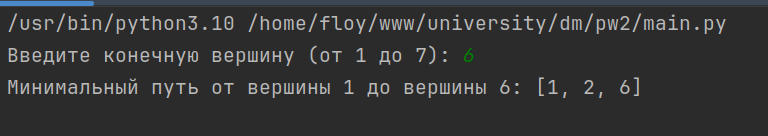
Цель работы: научится реализовывать алгоритм Дейкстры, понять, как находить кратчайшие пути от одной из вершин графа до всех остальных.

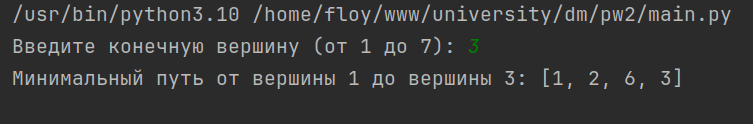
Содержание работы

Ход работы

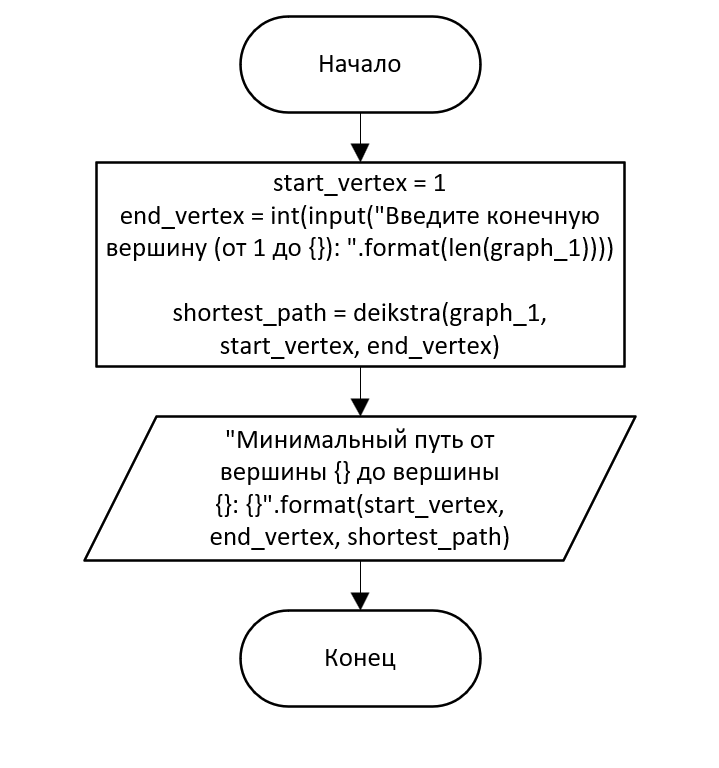
1. Реализовывал алгоритм Дейкстры, протестировал его, записал в отчет результаты.







Код программы:

Блок схема:  
  


import heapq

def deikstra(matrix, start\_vertex, end\_vertex):

matrix\_len = len(matrix)

start\_vertex -= 1

end\_vertex -= 1

distances = [float('inf')] \* matrix\_len

distances[start\_vertex] = 0

visited = [False] \* matrix\_len

shortest\_path = []

priority\_queue = [(0, start\_vertex)]

while priority\_queue:

\_, current\_vertex = heapq.heappop(priority\_queue)

if current\_vertex == end\_vertex:

break

if visited[current\_vertex]:

continue

visited[current\_vertex] = True

for neighbor in range(matrix\_len):

if matrix[current\_vertex][neighbor] != 0 and not visited[neighbor]:

new\_distance = distances[current\_vertex] + matrix[current\_vertex][neighbor]

if new\_distance < distances[neighbor]:

vertex = end\_vertex

while vertex != start\_vertex:

shortest\_path.append(vertex + 1)

for neighbor in range(matrix\_len):

if matrix[neighbor][vertex] != 0 and distances[vertex] == distances[neighbor] + matrix[neighbor][vertex]:

vertex = neighbor

break

shortest\_path.append(start\_vertex + 1)

shortest\_path.reverse()

return shortest\_path

graph\_1 = [

[0, 1, 0, 0, 0, 0, 1],

[1, 0, 0, 0, 0, 1, 1],

[0, 0, 0, 1, 1, 1, 0],

[0, 0, 1, 0, 1, 0, 0],

[0, 0, 1, 1, 0, 1, 0],

[0, 1, 1, 0, 1, 0, 1],

[1, 1, 0, 0, 0, 1, 0]

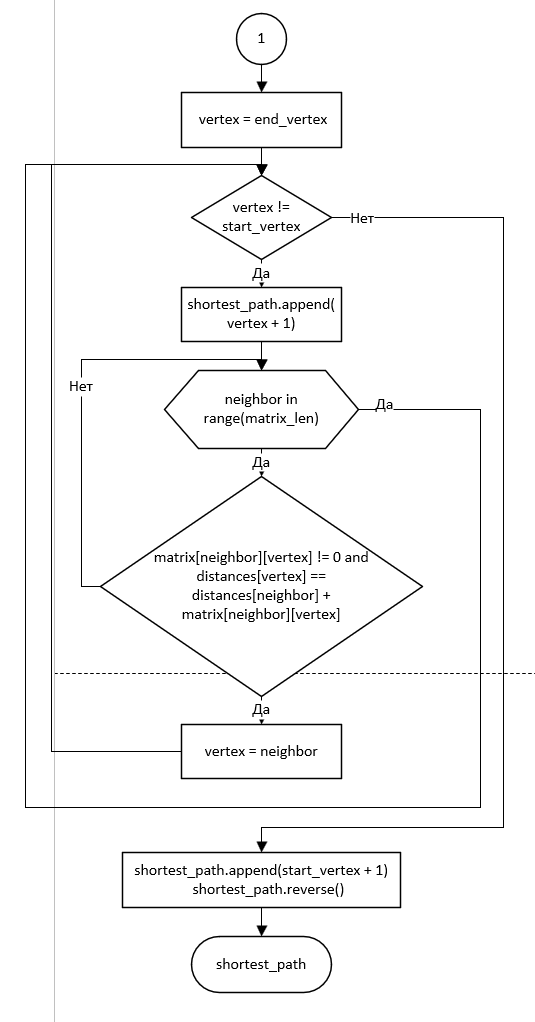
]

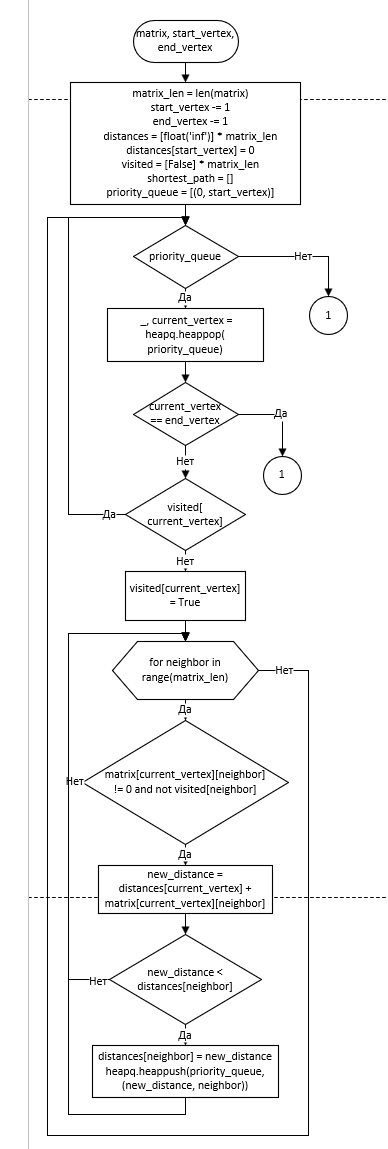
start\_vertex = 1

end\_vertex = int(input("Введите конечную вершину (от 1 до {}): ".format(len(graph\_1))))

shortest\_path = deikstra(graph\_1, start\_vertex, end\_vertex)

print("Минимальный путь от вершины {} до вершины {}: {}".format(start\_vertex, end\_vertex, shortest\_path))





**Вывод**: научился реализовывать алгоритм Дейкстры, понял, как находить кратчайшие пути от одной из вершин графа до всех остальных.