

**Computer Project 5 for****Discrete Mathematics & Mathematical Logic II course****Amangeldi Zhusubaliev, Kanykei Imanalieva and Zeqiye Zhan****American University of Central Asia**

**Table of Contents**

<b>1. Introduction</b>	<b>3</b>
<b>2. Text of Algorithm</b>	<b>4</b>
Flowchart	
P-code	
<b>3. Program Code</b>	<b>5</b>
<b>4. Exe-file</b>	<b>5</b>
<b>5. Results of Calculation</b>	<b>5</b>

## 1.Introduction

The aim of the project is to create an algorithm and the program for determining the shortest path between two vertices in a weighted graph.

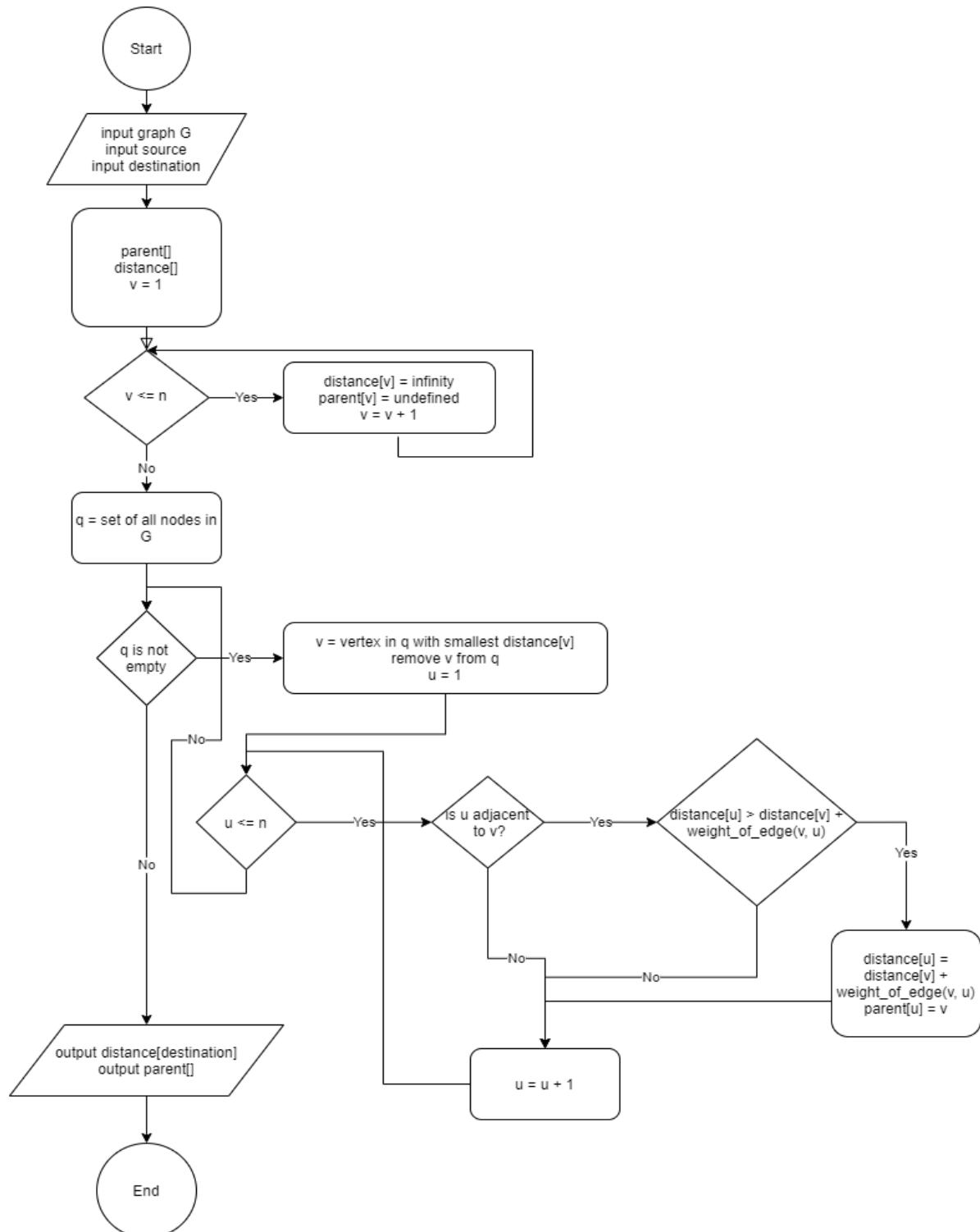
**weighted graph:** a graph with numbers assigned to its edges

**shortest-path problem:** the problem of determining the path in a weighted graph such that the sum of the weights of the edges in this path is a minimum over all paths between specified vertices

## PROJECT 5

## 2.Text of Algorithm

## Flowchart:



## PROJECT 5

**Pseudocode:**

```
function Dijkstra(graph, source, destination):  
    for each vertex v in graph:  
        distance[v] = infinity  
        parent[v] = undefined  
    distance[source] = 0  
    q = set of all nodes in graph  
    while q is not empty:  
        v = vertex in q with smallest distance[v]  
        remove v from q  
        for each adjacent vertex u to vertex v:  
            if distance[u] > distance[v] + weight_of_edge(v, u):  
                distance[u] = distance[v] + weight_of_edge(v, u)  
                parent[u] = v  
    return distance[destination], parent[]
```

**3. Program code & 4. Exe-file of working program**

**Please check the files “project.py” and “project.exe” uploaded together**

**<https://github.com/amangeldizhusubaliev/discrete2project5>**

## PROJECT 5

## 5. Results of Calculation

### First test problem:

Discription:  
Computer project of Discreate Math II in 2020 Spring:  
PROJECT 5: create algorithm to determine the shortest  
path between two vertices in a weighted grapgh  
By Zhan Zeqiye, Zhusubaliev Amangeldi, Imanalieva Kanykei

Test case: 1  
Shortest path from 2 to 3 has a total weights 5 and consists of 1  
vertex 2 to vertex 3, weight: 5

Test case: 2  
Shortest path from 1 to 3 has a total weights 4 and consists of 1  
vertex 1 to vertex 3, weight: 4

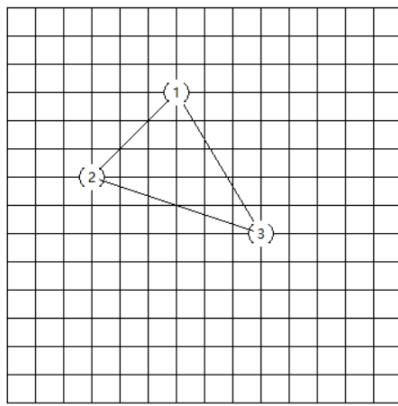
Test case: 3  
Shortest path from 1 to 2 has a total weights 3 and consists of 1  
vertex 1 to vertex 2, weight: 3

Project

Now you can choose two vertices and find the shortest path

Number of vertices:   Number of Edges:

First vertex:  Second vertex:  Weight of edge(1-999):



### Second test problem:

Discription:  
Computer project of Discreate Math II in 2020 Spring:  
PROJECT 5: create algorithm to determine the shortest  
path between two vertices in a weighted grapgh  
By Zhan Zeqiye, Zhusubaliev Amangeldi, Imanalieva Kanykei

Test case: 1  
Shortest path from 4 to 1 has a total weights 5 and consists of 2 edges:  
vertex 4 to vertex 2, weight: 3  
vertex 2 to vertex 1, weight: 2

Test case: 2  
Shortest path from 5 to 2 has a total weights 2 and consists of 2 edges:  
vertex 5 to vertex 3, weight: 1  
vertex 3 to vertex 2, weight: 1

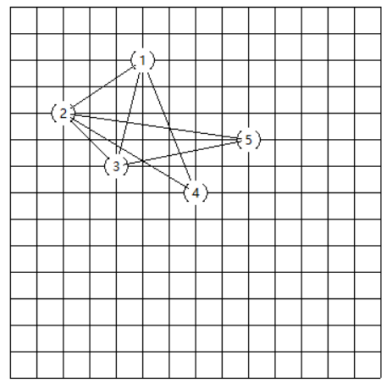
Test case: 3  
Shortest path from 5 to 1 has a total weights 4 and consists of 2 edges:  
vertex 5 to vertex 3, weight: 1  
vertex 3 to vertex 1, weight: 3

Project

Now you can choose two vertices and find the shortest path

Number of vertices:   Number of Edges:

First vertex:  Second vertex:  Weight of edge(1-999):



## PROJECT 5

## Third test problem:

Discription:  
Computer project of Discreate Math II in 2020 Spring:  
PROJECT 5: create algorithm to determine the shortest path between two vertices in a weighted graph  
By Zhan Zeqiye, Zhusubaliyev Amangeldi, Imanaliev Kanykei

Test case: 1  
Shortest path from 1 to 4 has a total weights 3 and consists of 3 edges:  
vertex 1 to vertex 2, weight: 1  
vertex 2 to vertex 3, weight: 1  
vertex 3 to vertex 4, weight: 1

Test case: 2  
There is no any path from vertex 6 to vertex 2

Test case: 3  
Shortest path from 5 to 2 has a total weights 3 and consists of 2 edges:  
vertex 5 to vertex 1, weight: 2  
vertex 1 to vertex 2, weight: 1

Project

Now you can choose two vertices and find the shortest path

Number of vertices:   Number of Edges:

First vertex:  Second vertex:  Weight of edge(1-999):