Homework 5 – Graph

TA: Patty (s221274155@gmail.com)
Deadline: 2018 Dec.11, 11:59pm

1. Objective

☐ Find the minimal cost spanning tree by using Kruskal's algorithm

☐ Find the minimal cost spanning tree by using Prim's algorithm

☐ Find the cost of the shortest path between two specified vertices by using Dijkstra's algorithm

2. Descriptions

You should use an **adjacency matrix** to implement the program.

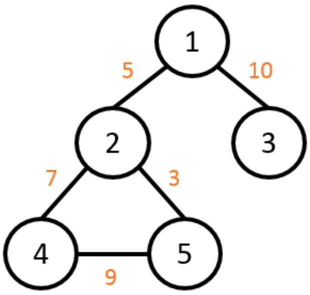
2.1 Kruskal's algorithm

- ➢ Function details:

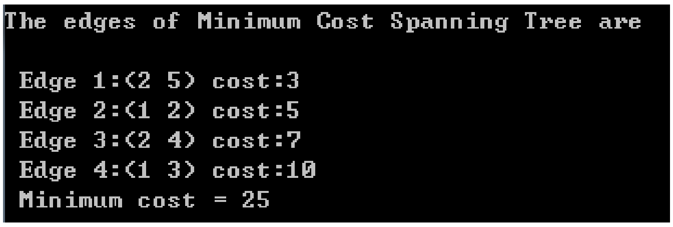
  Your program should read the standard input file that stores a graph and print out **each edge** of the minimum cost spanning tree according the processing order.

- ➢ Input format:
  - A text file includes the information of the weighted undirected graph.
  - The first line of the input file is *n* representing the number of vertices. The vertex No. starts from 1 and ends at n, that is, V={1, 2, …, n}.
  - The second line to the last line of the input file represent an adjacency matrix.
  - The following is a sample input file for the below graph:

```
5
0 5 10 0 0
5 0 0 7 3
10 0 0 0 0
0 7 0 0 9
0 3 0 9 0
```



- ➢ Output format:
  - The output consists of the starting node and the ending node of an edge with its corresponding cost.
  - The following is a sample output:



```
The edges of Minimum Cost Spanning Tree are

 Edge 1:(2 5) cost:3
 Edge 2:(1 2) cost:5
 Edge 3:(2 4) cost:7
 Edge 4:(1 3) cost:10
 Minimum cost = 25
```

2.2 Prim's algorithm

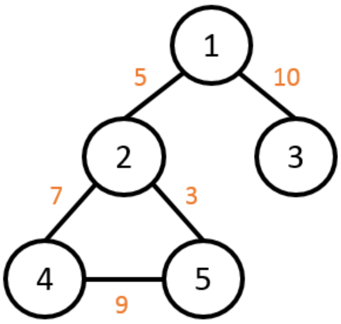
- Function details:

  Your program should read the standard input file and print out **each edge** of the minimum cost spanning tree according the processing order.

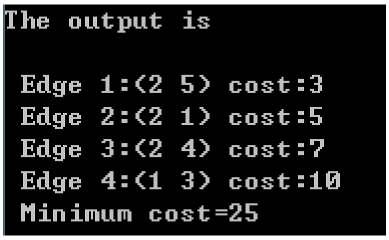
- Input format:
  - A text file includes the information of the weighted undirected graph.
  - The first line of the input file is *n* representing the number of vertices.
  - The second line to the (*n*+1) line of the input file represent an adjacency matrix. The vertex No. starts from 1 and ends at n, that is, V={1, 2, …, n}.
  - The last line is the specified vertex as the source vertex.
  - The following are two sample input files for the below graph:

```
5
0 5 10 0 0
5 0 0 7 3
10 0 0 0 0
0 7 0 0 9
0 3 0 9 0
2
```

```
5
0 5 10 0 0
5 0 0 7 3
10 0 0 0 0
0 7 0 0 9
0 3 0 9 0
3
```

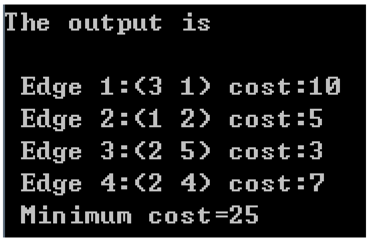


- Output format:
  - The output consists of the starting node and the ending node of an edge with its corresponding cost.
  - The following are the corresponding sample outputs:



```
The output is

 Edge 1:(2 5) cost:3
 Edge 2:(2 1) cost:5
 Edge 3:(2 4) cost:7
 Edge 4:(1 3) cost:10
 Minimum cost=25
```



```
The output is

 Edge 1:(3 1) cost:10
 Edge 2:(1 2) cost:5
 Edge 3:(2 5) cost:3
 Edge 4:(2 4) cost:7
 Minimum cost=25
```

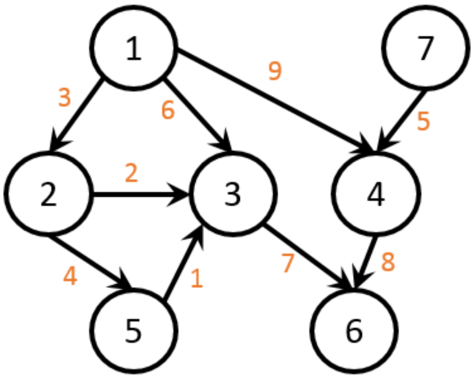
2.3 Dijkstra's algorithm

- Function details:

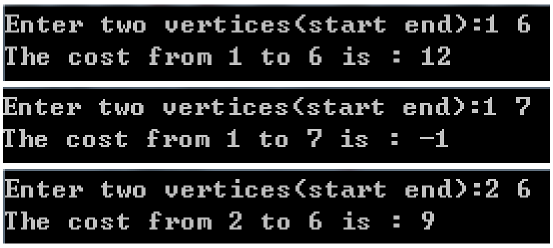
  Your program should read the standard input file and two variables from a user. Show the cost of the shortest path between two specified vertices in the weighted **directed** graph.

- Input format:
- A text file includes the information of the weighted undirected graph.
- The first line of the input file is *n* representing the number of vertices. The vertex No. starts from 1 and ends at n, that is, V={1, 2, …, n}.
- The second line to the last line of the input file represent an adjacency matrix.
- Read two variables from a user as the source and destination nodes, respectively.
- The following is a sample input file for the below graph:

```
7
0 3 6 9 0 0 0
0 0 2 0 4 0 0
0 0 0 0 0 7 0
0 0 0 0 0 8 0
0 0 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 5 0 0 0
```



- Output format:
  - Print the cost of the shortest path between the two vertices entered by a user. If the vertex can't reach another one, print -1.
  - The following is a sample output:



```
Enter two vertices(start end):1 6
The cost from 1 to 6 is : 12
```

```
Enter two vertices(start end):1 7
The cost from 1 to 7 is : -1
```

```
Enter two vertices(start end):2 6
The cost from 2 to 6 is : 9
```

3. Grade policies

10%- Readme file, code style, and comments in the source code

To keep source code maintainable and readable, you should **add English comments to your source code**. For this assignment, please also compose a small "**README.txt**" which contains a **brief** explanation of **how to compile your program** and **what problem you met**.

30%- Implement the Kruskal's algorithm function

See 2.1, and if you didn't use Kruskal's algorithm, you will get 0%.

30%- Implement the Prim's algorithm function

See 2.2, and if you didn't use Prim's algorithm, you will get 0%.

30%- Implement the Dijkstra's algorithm function

See 2.3, and if you didn't use Dijkstra's algorithm, you will get 0%.

Notice that if your homework is copied from your classmate, **you** and **your classmate** will get **0%** in this homework!

4. Summit

➢ Turn in

- System
  - To submit your files electronically, enter the following command from the csie workstation: turnin DS_I_2018.hw5 [your files…]
  - To check the files you turnin, enter the following command from the csie workstation: turnin –ls DS_I_2018
  - You can see other description about turnin from following link: https://www.cs.ccu.edu.tw/lab401/doku.php?id=turninhowto
  - This source code will be compiled and tested on the workstation
- Source code
  - Source code with appropriate comments
- Report
  - A document named "readme.txt". You should describe the details of your project in your readme file