

Assignment #4

Data Structures Deadline - 2022/12/23

Upload your assignment to Moodle before 11:59pm.

Please consult with TA if you have any questions.

TA's e-mail: p76114155@gs.ncku.edu.tw

Problem:

- (1) Implement Prim's algorithm to find minimal spanning tree.
- (2) Implement Dijkstra's algorithm to solve shortest path problem.

Input file description:

We will enter file name of the input file (xxx.txt), and the program needs to load the corresponding file.

The input consists of three parts. The first line is the numbers of edges (**n**). Then, **n** lines of edge will be entered in the format (**v1 v2 w**), **v1** and **v2** are the vertex of this edge, **w** is its weight, and there will have space to separate them. The last line of input is the starting point.

*The input format for both questions is the same.

*Every vertex will be represented by a lowercase English alphabet.

*You can check the attached files named **sample_p.txt** and **sample_d.txt** to see the full input for each question.

Output file description:

You must print the result in **console**. Additionally, you **must write it out to a file** named **output_p.txt** and **output_d.txt** for question (1) and question (2), separately.

The output format is described as follows:

(1) Prim's algorithm:

You need to output every edge found by Prim's algorithm.

For example, (b c 3) means edge (b, c) with weight 3.

*The b, c, 3 need to separate by space.

*The output should be sorted in **ascending order** according to the weight of each edge.

*The vertices on edges should also be sorted in **ascending order** according to alphabetical order.

*You can check the attached file **output_p.txt** to see the full output.

(2) Dijkstra's algorithm:

You need to output every vertex and its cost of arrival from starting point.

For example, (f 5) means that the cost of arrival to vertex f is 5.

* Noted that the starting point also need to be output with cost 0.

*The f, 5 need to separate by space.

*The output should be sorted in **ascending order** according to the cost of each vertex.

*You can check the attached file **output_d.txt** to see the full output.

Sample Input:

(1) Prim's algorithm:

```
11
a b 2
a d 9
b c 3
b d 8
b e 7
c e 4
d e 5
d f 10
e f 12
e g 11
f g 13
a
```

(2) Dijkstra's algorithm:

```
9
b a 4
a c 2
f c 9
d b 15
d f 5
d g 23
e b 17
e g 11
f g 13
d
```

Sample Output:

(1) Prim's algorithm:

```
a b 2
b c 3
c e 4
d e 5
d f 10
e g 11
```

(2) Dijkstra's algorithm:

```
d 0
f 5
c 14
b 15
g 18
a 19
```

Note:

You will need to turn in separate files named `prim.cpp` and `dijkstra.cpp` for each problem.

The input graph of **Prim's Algorithm** is a **undirected graph**, and the vertices of each edge will be sorted in ascending order according to **alphabetical order**.

The input graph of **Dijkstra's Algorithm** is a **directed graph**, each edge starts from `v1` to `v2`. For example, `(b a 4)` edge starts from vertex `b` to vertex `a` and its weight is `4`.

All the weights of edge are **positive integer**.

You are only allowed to use **C++** language for this homework assignment.

Besides, you need to explain what each function does and how to call it properly (i.e. what arguments should be passed) in the **Readme.txt**. Make sure your program can be executed in **Dev c++** and write code comments. Please compress all the files, and name the compressed file as **HW4_yourstudentID.rar** (or in **.zip** file) (e.g. **HW4_P76111111.rar**). Then upload the compressed file to the Moodle in time. Also, you shall print out the output in the `output.txt` after executing the `.exe`

The file structure should be like following figure:

```
| -HW4_P76*****.rar (.zip)
| | -HW4_P76***** (Folder)
| | | -prim.cpp
| | | -dijkstra.cpp
| | | -sample_p.txt
| | | -sample_d.txt
| | | -output_p.txt
| | | -output_d.txt
| | | -Readme.txt
```

Don't cheating, or you will get 0 for this assignment. If you can't finish the assignment before deadline, just hand in your unfinished code and report.