CSCE A331 Data Structures and Algorithms

Homework Assignment 4

|  |  |
| --- | --- |
| **First Name** | Quinton |
| **Last Name** | Odenthal |
| **ID#** | 31182634 |
| **Email Address** | quintonodenthal@gmail.com |

How to submit your Assignment

After filling all the parts in this file, please follow the following steps.

* Add your name and ID to the first page.
* Save the file in the original format (Docx or Doc)

(please do not convert to other file formats e.g. PDF, ZIP, RAR, …).

* Rename the file and include screenshots of your run results.

CSCE A311 *– HW4 – YOUR Last Name - YOUR First Name.docx*

**Example:** CSCE A311 *– HW3* - Smith - John.docx

* For source code:

Zip up the source code for each question, and rename your zip file as

*YOUR Last Name - YOUR First Name-HW4-Question1.zip*

* Upload all your files and submit homework (only using Blackboard)

Assignment

**No collaborations are permitted, copying classmates work and submitting as your own, or seeking assistance of tutors or others is considered Plagiarism.**

* The map below represents a crude graph of places around Anchorage. Each location is a node. The red lines indicate edges between nodes. The edges have weights, which is the distance between the nodes. All of this information is stored in the data below, which you can save in a text file and read into your program.



The format of the file is:

Line 1: Number of nodes in the file

The following then repeats for the number of nodes in the file:

Number of node (e.g. 1 on the map)  
Name of node (e.g. "JBER" on the map)  
Number of edges from this node  
NeighboringNodeNumber  WeightOfEdgeToNode  
... (repeats for the number of edges)

In the data below, Tikahtnu is node number 2 and has 4 edges. It is connected to Node 1 with weight 2.71, Node 3 with weight 2.21, Node 8 with weight 1.8, and Node 9 with weight 2.39

15

1

JBER

1

2 2.71

2

Tikahtnu

4

1 2.71

3 2.21

8 1.80

9 2.39

3

Science & Nature Museum

5

2 2.21

5 1.19

7 2.60

8 2.18

13 2.40

4

Sullivan Arena

4

5 1.31

6 0.79

14 1.25

15 0.76

5

Merrill Field

3

3 1.19

4 1.31

14 1.45

6

Anchorage Museum

3

4 0.79

7 0.41

15 1.58

7

AK Railroad

2

3 2.60

6 0.41

8

Cheney Lake

4

2 1.80

3 2.18

9 1.25

13 2.09

9

Totem 8

4

2 2.39

8 1.25

11 1.89

13 2.90

10

Campbell Creek Science Center

1

11 1.61

11

Crime Lab

4

9 1.89

10 1.61

12 1.89

13 1.38

12

Golden Donuts

3

11 1.89

13 0.94

15 1.40

13

UAA

6

3 2.40

8 2.09

9 2.90

11 1.38

12 0.94

14 0.81

14

Don Jose

4

4 1.25

5 1.45

13 0.81

15 1.02

15

Steamdot Coffee

4

4 0.76

6 1.58

12 1.40

14 1.02

To do:  Read this data into a graph.  Let the user enter a source and target destination (by number is OK). Then run Dijkstra's algorithm to find the shortest path between the source and target, and output the total distance and path that should be traveled to reach the destination from the source.  Prompt the user if he or she wishes to repeat the calculation.

|  |
| --- |
| The run result |
|  |

* Using the same graph as problem 1, write a program that calculates a Minimum Spanning Tree and outputs the edges and the total edge weight of the MST. The MST could be used if you wanted to connect all the nodes in some way (e.g. water, electrical network connection). Your program can use any MST algorithm that you like as well as the data structures of your choice for internal implementation (e.g. making a set or queue). You should be able to reuse a large amount of your code from the previous problem to write this one.

|  |
| --- |
| The run result |
|  |