

COSC 2P03 Advanced Data Structures: Assignment 4

Instructor: Yifeng Li^{*1}

¹Department of Computer Science, Brock University

November 28, 2021

1 Introduction to the Problem

Transportation network is a model of the real-life transportation system. It is a typical example in graph theory research and plays an important role in a variety of real-life applications (e.g. planning and navigation). In this assignment, you are given a simplified version of Ontario's land transportation network (see Figure 1), and need to use graph algorithms to solve a few tasks.

2 Your Tasks (8 Marks)

1. Define a class named **TransportNet** which has data attributes to represent the graph (i.e. the transportation network) and has methods as required below.
2. Within the **TransportNet** class, define a method named **readData** to read the information from the two attached CSV text files (**cities.txt** for vertices in your graph, **distances** for edges in your graph). These two files will help you create your weighted undirected graph. You should use adjacency matrix to represent edges. (1 mark)
3. Within the **TransportNet** class, define a method named **findShortestPath** based on Dijkstra's algorithm to find the shortest path between two given cities on the graph. For example, if you call **findShortestPath("Windsor", "Ottawa")**, it should return the shortest path (i.e. a sequence of city names) and the total cost (i.e. the sum of weights on the shortest path). You can either use Java's built-in priority queue implementation or implement it yourself. (2 marks)
4. Within the **TransportNet** class, define a method named **MST** based on Kruskal's idea to return a minimum spanning tree of your graph and the total cost of this tree (i.e. the sum of weights on the tree). You are allowed to use code from the textbook and lecture slides. (3 marks)
5. In the **main** function, create an instance/object (with name **tn**) of your **TransportNet** class. Call **readData(<your data path>)** to create your graph. Call functions **findShortestPath("Windsor", "Ottawa")** and **findShortestPath("Arnprior", "NOTL")**. After each call of this function, you should write your results into a text file named **results.txt**. Use file IO rather than copy-and-paste printed messages from the screen. Furthermore, call **MST()** and then write the returned results to the **results.txt** file. (1 mark)

Program comments: to maximize readability, you should properly comment your program. (1 mark)

3 Submission

- Your source code.
- A PDF printout of your source code.

^{*}E-mail address: yli2@brocku.ca

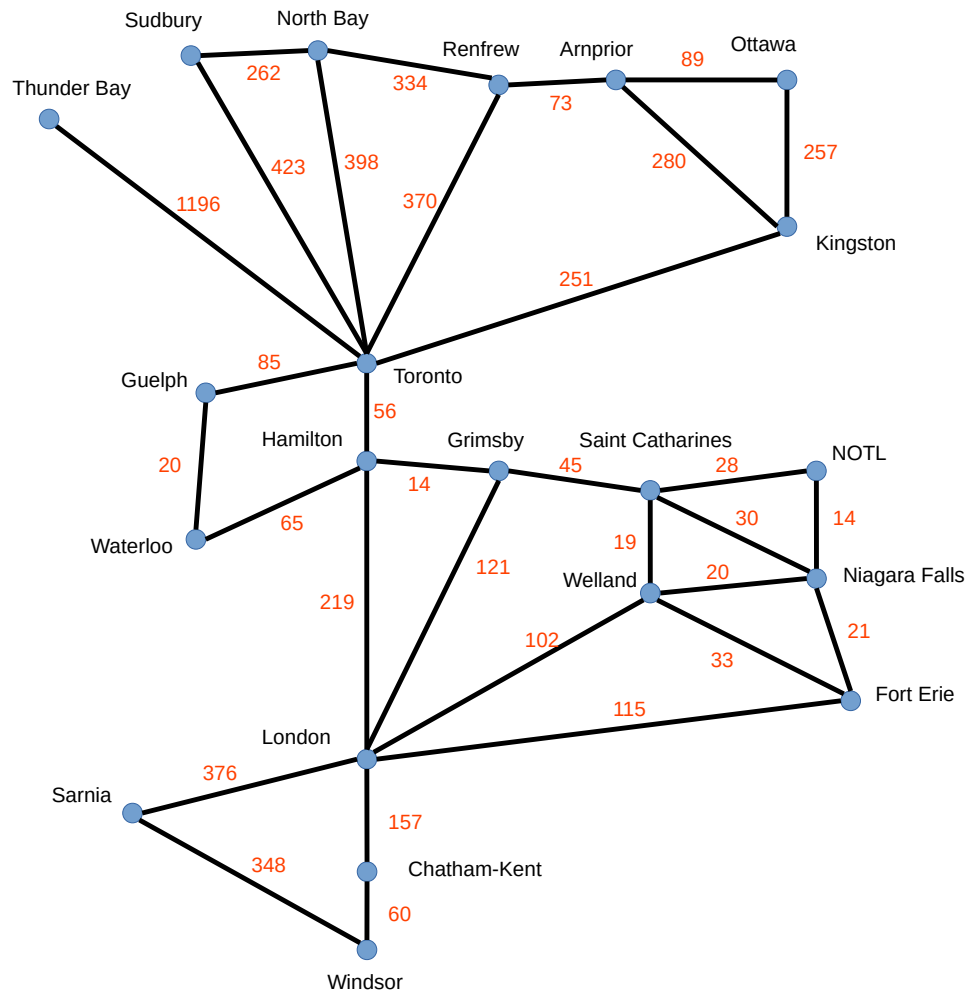


Figure 1: A simplified transportation network for Ontario's land transportation system.

- The text file `results.txt` that shows all saved results as required above.
- Compress the above files in a zipped folder named `COSC2P03_A4_YourFirstname_YourLastname_StudentNumber.zip` and submit it through Sakai before indicated due time.
- Late submissions will not be accepted.
- **Note: you should submit both your Java source code and your PDF printout. Missing any of them will result in a zero grade for this assignment.**

4 Academic Integrity

This assignment should be tackled individually. Outsourcing or teamwork is not allowed. Violation of this requirements will be seriously processed in accordance with university policies.