Lab Assignment 6: Graphs

1 Introduction

The purpose of this lab is to practice working with graph data structure, but also to explore the connections between different data structures as well as the relationship between data structures and algorithms.

The task is to calculate a solution for a practical problem: determing the shortest path to travel from one city to other cities. The road map (see Fig 1) shows the roads that directly connect the cities. Each road has a weight representing the distance of the road.

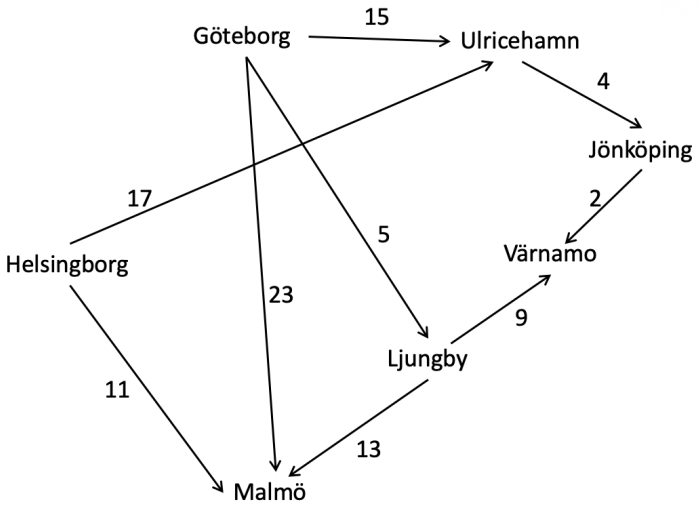

Figure 1: The Road Map

A guide to implement the program is given as below:

- Represent the road map in a weighted directed graph. The cities are represented as vertices (nodes) in the graph and the roads are represented as the edges connecting the vertices.
- Apply Dijkstra's algorithm over the graph to calculate the distances of shortest paths between the cities.
- To make it more efficient, use a min binary heap to organize the priority queue when implement Dijkstra's algorithm.

2 Requirements

You should implement the interfaces of graph ADT and heap ADT as defined in the header files and a main function (included in dijkstra_supplied_codes.zip). As defined in the header

file, graph is represented using **adjacency list**. You can adapt the program implemented in the exercise for the adjacency list representation of graph. Priority queue is implemented using **min binary heap**. Moreover, there is a function, called `decreaseKey` in the queue ADT. This function is to decrease the key value of a node and move up the node until the min heap property is satisfied. As defined in the `main.c` file, the function *dijkstra* is to calculate the distances of shortest paths between cities. Once you've got your program implemented correctly, you should be able to produce output like the example shown in Fig 2, where the input from user is 3 (i.e. the starting city is Göteborg).

```
0: Jönköping, 1: Ulricehamn, 2: Värnamo, 3: Göteborg, 4: Helsingborg, 5: Ljunby, 6: Malmö
Enter the city :  3

The distance of the shortest path for travelling from Göteborg to
Jönköping is 19
Ulricehamn is 15
Värnamo is 14
Göteborg is 0
Helsingborg !!! no connection between these two cities
Ljungby is 5
Malmö is 18
```

Figure 2: example program output

Deliverables

The deliverable is a zip file in which the files are organized as in Fig 3. The provided header files must be included in your implementation. Variables in your code files should have proper names, and the code has to include sufficient comments for readability.

The source code has to be thoroughly tested as well as to adhere to the recommendations on *Standard C and portability*, which you can find in Canvas. You can compile your code using more strict flags `-Wall` together with `-pedantic` before submitting the code, e.g.

```
$ gcc -Wall -pedantic -I./include src/*.c -o printer_simulation
```

Please include a comment in your program (all `.c` files) with the year and your name to indicate authorship:

```
//2025 Your Name
```

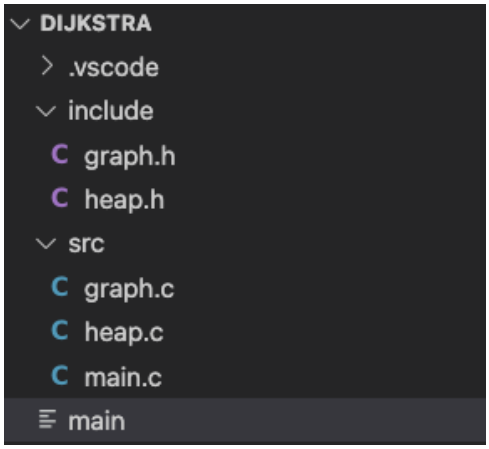

Figure 3: file organization