CS202, Spring 2024

Homework 4 - Graphs

Due: 14/05/2024

Before you start your homework, please read the following instructions carefully:

FAILURE TO FULFILL ANY OF THE FOLLOWING REQUIREMENTS WILL RESULT IN A GRADE SCORE OF 0 (zero) WITHOUT ANY CHANCE OF REDEMPTION.

- See the course page for any late submission policies and Honor Code for Assignments.
- Upload your solutions in a **single ZIP archive** using the Moodle submission form. Name the file as `studentID_name_surname_hw4.zip`.
- Your ZIP archive should contain **only** the following files:
    - Your `.cpp`, `.h`, `.pdf files`, and **Makefile (No Makefile results in 50% deduction in points)**.
    - Do not forget to put your name, student ID, and section number in all of these files. Add a header (see below) to the beginning of each file:

      ```
      /*
       * Author : Name & Surname
       * ID: 12345678
       * Section : 1
       * Homework : 4
       * Description : description of your code
       */
      ```

    - Do not add unnecessary files, such as the auxiliary files generated from your preferred IDE.
- Your code must be **compilable**.
- Your code must be **complete**.
- You ARE NOT ALLOWED to use any data structure or algorithm-related function from the C++ standard template library (STL) or any other external libraries.
- Your code must contain the proper **comments** needed to understand its function.
- Your code must run on the **dijkstra.cs.bilkent.edu.tr** server.
- Answers to Q1 and Q2 **MUST** be hand-written. You may either write on a piece of paper and then scan it or use a tablet and export a PDF of your hand-written answers.
- The output of Q3 **MUST** match the format shown in the output of the example code.
- For any questions related to the homework, contact your TA: *can.bagirgan@bilkent.edu.tr*

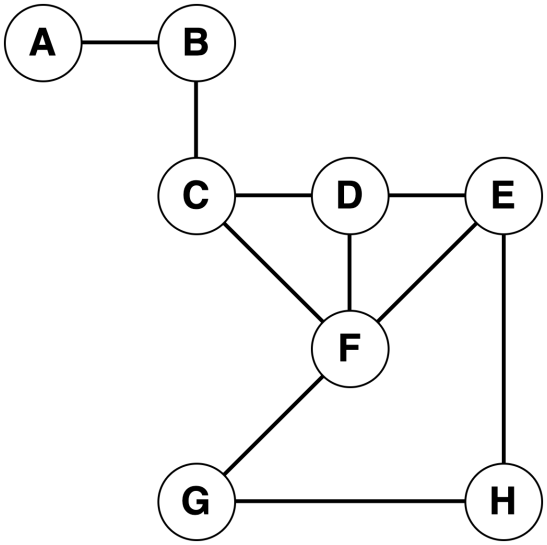
DO NOT START YOUR HOMEWORK BEFORE READING THIS SECTION CAREFULLY! THE INSTRUCTIONS MAY DIFFER FROM HOMEWORK TO HOMEWORK!

Question-1 (20 points)

For the graph given below, give the sequence of vertices when they are traversed starting from *vertex C* using

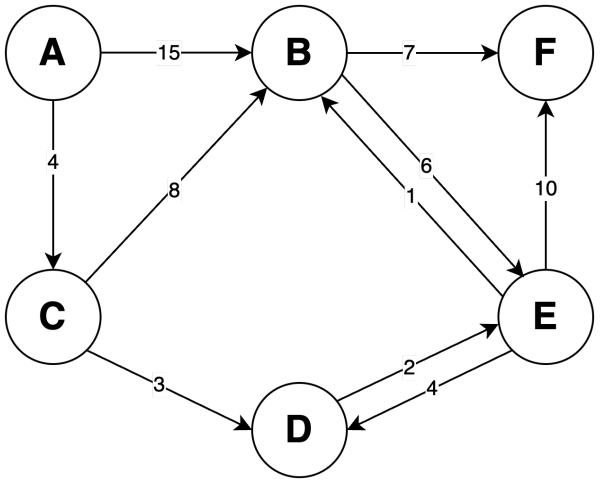
- The depth-first traversal algorithm
- The breadth-first traversal algorithm

In your solution, for a vertex, use the lexicographical order to visit its neighbors.



Question-2 (10 points)

For the following weighted directed graph, find the shortest paths from *vertex A* to all other vertices using Dijkstra's shortest path algorithm. Show all steps of Dijkstra's algorithm.



Question-3 (70 points)

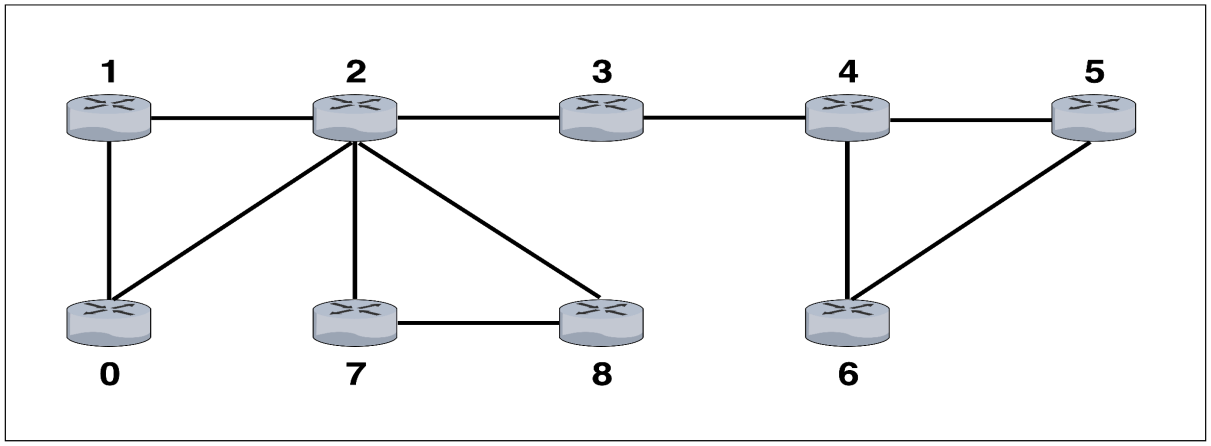
Suppose that a networking company wants to use a system that organizes the connections of its routers. In this programming assignment, you will implement a part of this system. In your implementation, you are supposed to compute the minimum number of routers that a packet should visit to reach its destination router and the diameter of the graph. (***Diameter***: *The longest of the shortest paths*)

In this implementation, you will be given a map file, which stores the information about the routers and their connectivities (cables). The first row of the input file indicates the number of the routers and each subsequent row includes information of a particular router. This information contains <router id> <degree> <neighbor id>* tokens where there might be many neighbor IDs. For a particular router K,

- **Router ID:** ID of router K.
- **Degree:** is the number of the neighbors of router K (i.e., it is the number of routers that have a direct cable from/to router K)
- **Neighbor ID:** ID of the adjacent router

For instance, the following table gives an example input file. This file contains 9 routers, as indicated in the first line. After this first line, there exists information for each router. For example, the fifth line of this file indicates that the router has an ID of 3 and has a direct road to two other routers (i.e., it has two neighbors). The IDs of these neighbor routers are 2 and 4. The graph given in this example input file is illustrated in the figure given below.

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



In this assignment, your program will take three command line arguments. The first one is the name of the input file and the other two are the names of a source router and a destination router. Your program will output 2 things. First, the minimum number of routers to be visited to deliver a packet from the source router to the destination router. (It will also output the IDs of the routers in the visiting order.) Second is the diameter[1] of the graph.

The name of your executable must be **pathFinder**.

```
username@dijkstra:~>./pathFinder <inputFile> <sourceRouterId> <destRouterId>
```

This command calls the executable with three command line arguments that are the name of the input file, the source router, and the destination router.

For the example input file given above (assume that its name is “input.txt”), the sample outputs are given below:

```
username@dijkstra:~>./pathFinder input.txt 3 5
Path consists 3 routers:
3
4
5
Diameter of the graph is 4

username@dijkstra:~>./pathFinder input.txt 0 6
Path consists of 5 routers:
0
2
3
4
6
Diameter of the graph is 4
```

[1] *The diameter of a connected component (in this case the graph itself since the graph includes only 1 connected component) is the longest of the shortest paths between any pair of vertices within this connected component.*

In your implementation, you should make the following assumptions:

- The cables between routers are bidirectional. That is, if router 0 is reachable from router 1, then router 1 is also reachable from router 0.
- Since the real distances between routers are not given, you should assume that all distances are equal to each other.
- There may exist different solutions, but the minimum number of routers to be visited should always be the same. These solutions may contain different lists of the visited routers. It is sufficient to output one of such solutions.
- You may assume that the input file is always valid. The IDs of the routers are between 0 and N-1, where N is the number of routers. The data in the input file are separated with a white space character.
- You may assume that all routers are connected. (No isolated routers)
- Please do your implementations wisely. Think about edge cases and large-sized inputs.

Good luck to you all 🙂