COLORADO MESA UNIVERSITY

Code Task 2 Due: 11:59 PM Mar 31st, 2025 20 Points

CSCI 250 Introduction to Algorithms

Graph Theory Algorithms

By: McKenzie Swindler, Stephen Bruner, Russell Cannon, Patrick Murphy

Task Description

Construct a data input file containing the representation of the graph pictured below a "DIGRAPH". The "DIGRAPH" vertices labels are the integer in each node circle. Arrows indicate edge directions. Edges without arrowheads are bidirectional and exist as two directed edges. Your program is to accept a single command line argument when invoked being the data file name. Your program will then read the file contents into memory (using your designed dynamic memory-based data storage structure). It would be unwise to code your data structure based on the assumption that the graph size is limited to what is pictured below. The test/demo graph may be larger (have extra nodes). Once the program reads the file data, close it, you may not read from it again. Your program will then prompt the user for three source vertices and destinations. Sample interaction:

Provide 3 sources: 7 19 26
Destination 1?: 32
Shortest Path 1 from vertex "Number":
Shortest Path 2 from vertex "Number":
Shortest Path 3 from vertex "Number":

Destination 2?: 22
Shortest Path 1 from vertex "Number":
Shortest Path 2 from vertex "Number":
Shortest Path 3 from vertex "Number":

Destination 3?: 3
Shortest Path 1 from vertex "Number":
Shortest Path 2 from vertex "Number":
Shortest Path 3 from vertex "Number":

Program Termination

Upon a negative value user input the program will terminate, to do so it must:

- Write a properly formatted log file of all interaction from invocation to termination
- For each Shortest path in the log it should also present running time of the calculation in nanoseconds
- De-allocate all dynamically allocated memory
- Record the overall runtime in seconds
- Record the date and time of the run
- Display a termination message on the screen

Our Program

We decided to use the Breadth First algorithm to complete this task. Between BFS and DFS, BFS is best suitable for finding the shortest path between nodes.

Blacklist

- C++ code using STL constructs or libraries.
- Any data structure not of your making.

Deliverables:

- C++ code implementing the Union-Find data structure and solving the problem.
- Classes should be coded using interface and implementations in separate

  les i.e., ḣ" andċpp". Templated classes should be prototyped and implemented in their interface ".h"

  les. required).
- A LATEX based PDF document explaining your algorithm's time complexity analysis and any optional features you implemented.