

Jerry Huang
Period 2
APCS
Kuszmaul

Dijkstra's Algorithm Readme

I: Introduction:

In this Dijkstra's assignment, we were asked to implement a class called Dijkstra that has a constructor method that takes a HashMap that maps a String to a Hashmap that maps Strings to Integers, which represents an adjacency list representation of a graph, and a String that represents a starting node, and produces public variables called path of type HashMap (from String to String) that represents the sequence of vertices corresponding to the single source shortest path as discovered by Dijkstra's algorithm. In addition, we also had to implement an Interface that that requires the method ArrayList next(String, Dijkstra) - the return value representing the shortest path between the nodes represented by the argument string and the start node within the graph implicitly represented by the argument to the constructor of the Dijkstra object.

II: How This Fulfills the Specification

This project fulfills the specification for the Dijkstra's assignment. It can perform Dijkstra's algorithm on a graph of nodes and prints the results in a formatted table. It will also find the shortest path available from the starting node to the destination node.

III: Explanation of Current Errors

Currently, the program has one minor error. If the user wants to find the path between two nodes that cannot be reached, the program will crash and an error will be produced. Unfortunately, I did not have enough time to fix this bug but everything else in the program works properly and as specified in the assignment.

IV: Overview of Code

My program is split into three main classes. The Main class is used to first create the graph, then call the Dijkstra constructor. The Dijkstra class performs the actual Dijkstra's algorithm and prints out a table. The Shorty class finds the shortest path between two specified nodes. The Main class is not encapsulated very well, which is a mistake on my part that will most likely not happen again in the next assignment. The purpose of the Main class is to create all the node adjacencies and put them into HashMaps so that when the Dijkstra constructor is called, the program can go straight into performing Dijkstra's algorithm. The Main class also prints out the node adjacencies for testing and verifying purposes to make sure that the graph was created properly. The Dijkstra class is where most of the work happens. The constructor takes in the

node adjacencies HashMap, and the start node. The initializeVariables() method simply initializes all the necessary arrays like nodes, k, dv, and prev. These arrays will be used to keep track and update the information in the table for when it is printed at the end of the algorithm. The algorithm starts in the runDijkstra() method. It runs through a while loop, which will not stop until all the nodes on the map have been selected. It iterates through all the nodes adjacent to the current selected node, and adds all the distances to a priority queue. It then calls the selectMinDistance() method, which retrieves and removes the smallest value out of the priority queue. The name of the node with the shortest distance is then returned back to the runDijkstra() method. This node then becomes the new selected node, and the method now iterates through all the nodes adjacent to the new selected node. This loop continues until all nodes have been selected and the shortest distances have all been updated. The printResults() method is called last and prints a table showing the results.

V: Major Challenges

Major challenges when coding this program was correctly coding the logic in the runDijkstra() method. I found that simply adding a while loop or changing the position of one line of code could make the difference between a working and not working program.

VI: Acknowledgements

Special thanks to Mr. Kuszmaul for demonstrating some code in class which really helped give me a kick start on the program. He taught me how to create HashMaps and how we could use them in this program. Having this clear understanding of HashMaps and what can be done with them in Java benefitted me a lot when creating the runDijkstra() method. Also, thanks to the Java documentation which I used to learn the methods associated with priority queues.