

DATA STRUCTURES AND ALGORITHMS

PROJECT REPORT

MINI PROJECT 2



Group Members Name & Reg #:	<u>Muhammad Haris Irfan</u> (FA18-BCE-090)
Class	Data Structures and Algorithms CSC211 (BCE-3B)
Instructor's Name	Dr. Omer Ahmed

Mini Project 2: Finding Shortest Path using Dijkstra's Algorithm.

Purpose

To Implement the Dijkstra's Algorithm which is a Breadth First Search (BFS) algorithm for finding shortest path from starting vertex (src) to every other vertex in the graph.

Project Description

- In this Project, **if / else** Statements, **Arrays, Pointers, Structures** and **Functions** have been used.
- Initially we are given an integer two dimensional array named 'graph' in which value of weights are added correctly on their respective positions, and a 'distance_array' pointer that points to an Array that will have all the distance corresponding to the source node, initially the all the value of 'distance_array' are set to -1, here '-1' corresponds to infinity.
- I have globally declared an integer Array named 'Visited' of size equal to the total number of vertices.
- In my code I have used some global variables as well, such as 'int carry' and 'int smallest', 'carry' is the base cost from a node being explored to the source node, and 'smallest' will have the smallest path cost to decide the next traversals, initially I have set it equal to a large value.

- I have also globally declared an integer named 'value' and set its value equal to 0, when the code is run the first time 'if' condition checks the value of this variable and as it's the first time and value is 0, the first part of the code runs, at the end of this part and before calling the next value for traversal the value is set to 1, so that this part of code doesn't run again or is skipped.
- After the first time the smallest path cost is searched, whose node is not yet visited, nor its value is -1 or 0.
- When the function is called again with a smaller valued node, it again checks the previous values and if the path cost already entered needs to be updated (i.e. $\text{new cost} < \text{old cost}$) it updates it, or a value that is not already added is updated, when the two values compared are equal it does nothing.
- This Program keeps calling itself recursively with different values until all the node values are in the 'Visited' array and then the 'Shortest Path Finder' Function terminates after which the distance array is printed, with all the shortest path from the source node with source node having a value of 0.
- I have also given the user the choice to decide the source node on runtime, for which I have taken the value from user by prompting and asking for a value from 0-9(because our source nodes are from 0-9).

Problems faced while Implementation

Initially the project looked difficult, but with the help of debugger and dry-runs I didn't face any major problem.

Limitations of Code

The Code asks the user to enter a value of source node from 0-9 but if the user enters a negative value or a value greater than 9, then the output is wrong, because our nodes are from 0-9 and cannot be any other number.

***Working Program file is also inside the folder.**

THE END