PROJECT ABSTRACT

Advanced Data structures Laboratory

SHORTEST PATH FINDER IN A MAP

Swaminathan Navinashok

2019115126

The aim of this project is to develop an application that could find a shortest path in a user input text file (typically a hardcoded and formatted map-like structure) through show casing

CONCEPTS USED

1. OOPS concepts

2. Graph algorithms

   i. Breadth first search
   ii. Depth First Search
   iii. Dijkstra's algorithm

3. STL
   i. Vectors
   ii. Priority queues
   iii. Stacks
   iv. Queues

The project finds the shortest path between two nodes of user choice through an algorithm of user choice and prints the path to follow (and also the iterations of the algorithm for verification purposes)

Classes used

Location :

An Object of type Location for every entry in the input matrix is preserved

Every such object has:

i. name
ii. coordinate in the matrix
iii. parent's coordinates in the matrix
iv. whether or not it has a neighbor at left, right , above or below
v. whether or not the node has been visited during the search algorithm
vi. set and get functions for the above
vii. overloaded << operator

Map :

An object of type map has/stores :

i. a 2d vector of location objects called locations
ii. a user input start and end vertex location objects and their cordinates
iii. whether or not a map has been loaded
iv. whether or not a path has been found
v. a constructor that takes a filename as input and initializes the locations 2d vector
vi. Set and get functions for the above
vii. Overloaded () operator that returns a location object at the coordinates set by the arguments

Functions Used

printMap :

Prints the map

PrintCoordinates :

Prints the path to go from start to end vertex by mentioning the coordinates of each step of the path

Success:

Sets a path trailer from start to end if path was found

SearchPath:

Templated function to search for path in BFS as well as DFS using a queue or stack of locations as a template parameter and calls **Success** if a path is found

searchDjikstra:

Searches for a path using dijiksta’s algorithm and calls successDjikstra if successful

successDjikstra:

Sets a path trailer from start to end if path was found and computes total cost for traversal

optionsMenu:

To print menu and input choice