GTU Department of Computer Engineering
CSE 222 - Spring 2023
Homework 8 Report

BEYZA ACAR

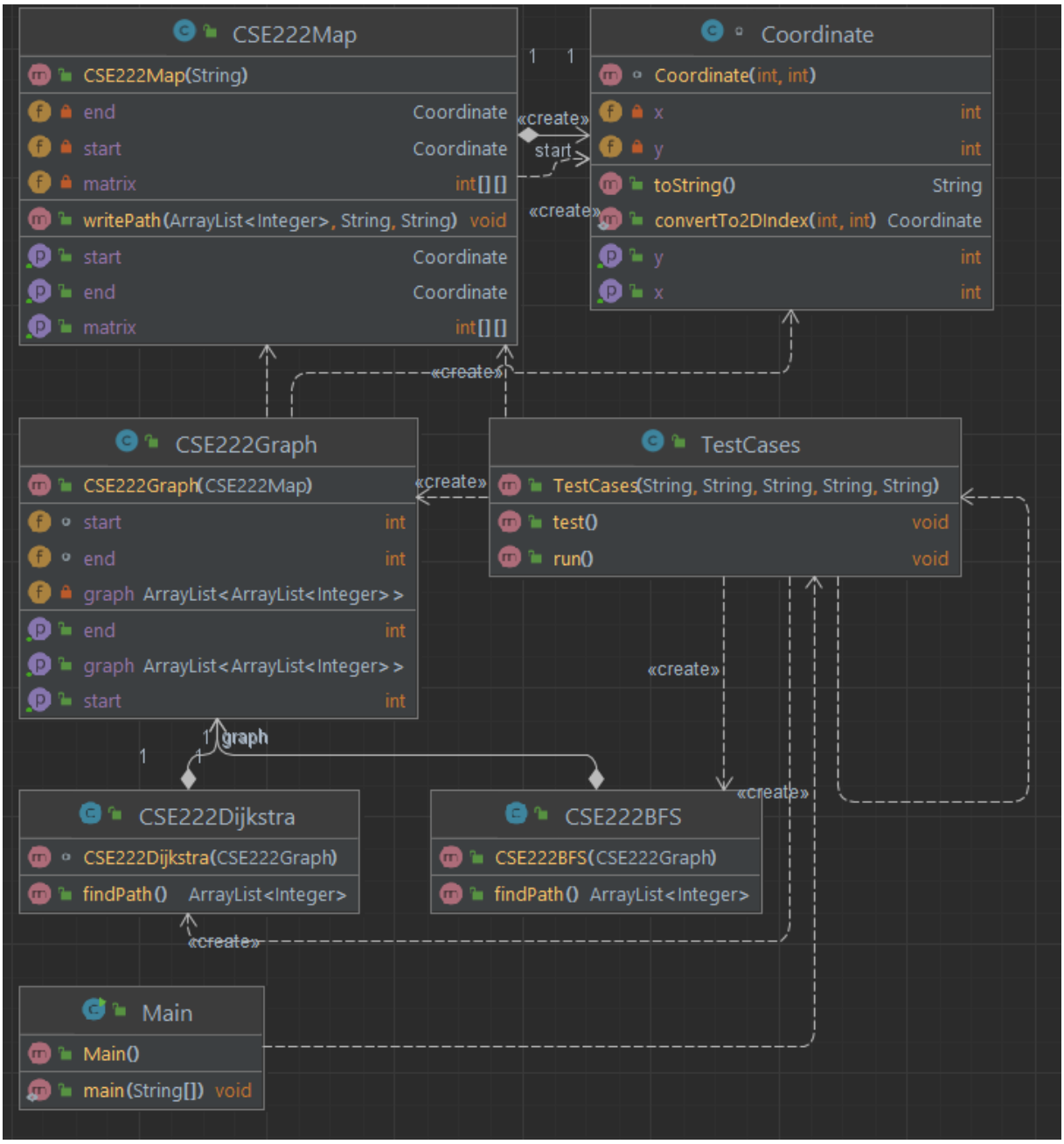
200104004065

04.06.2023

1- ***Problem Definition***

The path planning problem is the task of finding an optimal or feasible path from a starting point to a goal point in each map while avoiding obstacles. The optimal path is the shortest path between the starting point and the end point on the map. In the homework, you are responsible for finding feasible path(s) on the maps. If you can find the optimal path(s), you will get extra points for each map.

2- ***Uml Diagram***



3-Problem Solution Approach

Firstly, I created the adjacency list representation of the graph. The graph is represented as an ArrayList of ArrayLists, where each ArrayList represents a node and contains its neighboring nodes. In order to minimize time complexity, I didn't skip any nodes and used an ArrayList, which increased the memory complexity. To reduce memory complexity, I used linear indices instead of keeping track of two coordinates throughout the code. For example, in a 2D array, the linear array is formed by concatenating the rows. Indexes are found using the formula: i * matrix.length + j (where i is the row number and j is the column number).

EXAMPLE OF LINEARIZATION:

* 2D ARRAY : 1 2 3 LINEAR ARRAY : 1 2 3 4 5 6 7 8 9
* 4 5 6
* 7 8 9

For instance, the index of 5 in a 3x3 matrix is 1 * 3 + 1 = 4.

This approach helped reduce memory complexity a little throughout the code.

In the BFS algorithm, I used the classic BFS algorithm with a queue.

In the Dijkstra's algorithm, I didn't use a priority queue, resulting in a time complexity of O(V^2) (where V is the number of vertices or nodes in the map).

4-Time Complexities

a.Dijkstra algorithm

As mentioned above, since I didn't use a priority queue in the Dijkstra's algorithm, my time complexity became O(V^2) (where V is the number of vertices or nodes in the map).

b.BFS algorithm

The time complexity of my BFS algorithm is O(V), where V represents the number of elements in the map, the number of vertices, or the number of nodes. It starts from the "start" node and checks the next "neighbor". If the number of neighbors was unknown, the complexity would be O(N), where N represents the number of neighbors. However, since the maximum number of neighbors is 8, the complexity simplifies to O(1). Therefore, our overall BFS path-finding algorithm has a time complexity of O(V).

5-OUTPUT FILES

There will be four types of output files. The first two files will be PNG files containing visual representations of the paths found by the Dijkstra and BFS algorithms. The remaining two files will be TXT files that contain the coordinates of the paths found by the Dijkstra and BFS algorithms.