CENG317 – Principles of Artificial Intelligence

Homework #1

Due: 21.10.2023 @23:59

ATTENTION:

- Programs which contain **errors** will **NOT** be graded.
- In all of the programs, use **COMMENTS** where necessary.
- Code blocks inside comments will **NOT** be graded.
- Send only .text or .java files and the .pdf file in a .zip file named as **"hw1_yourName_yourID.zip"**.

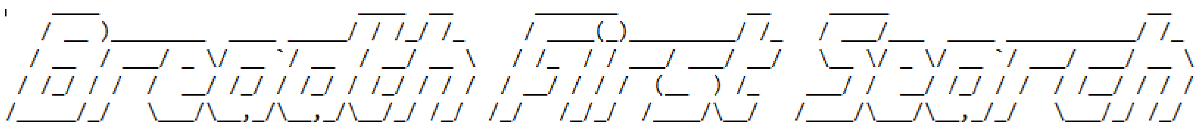
Disciplinary action to be taken against cheating is arranged by the Rules and Regulations Governing Student Disciplinary Actions in Institutions of Higher Education.

QUESTIONS

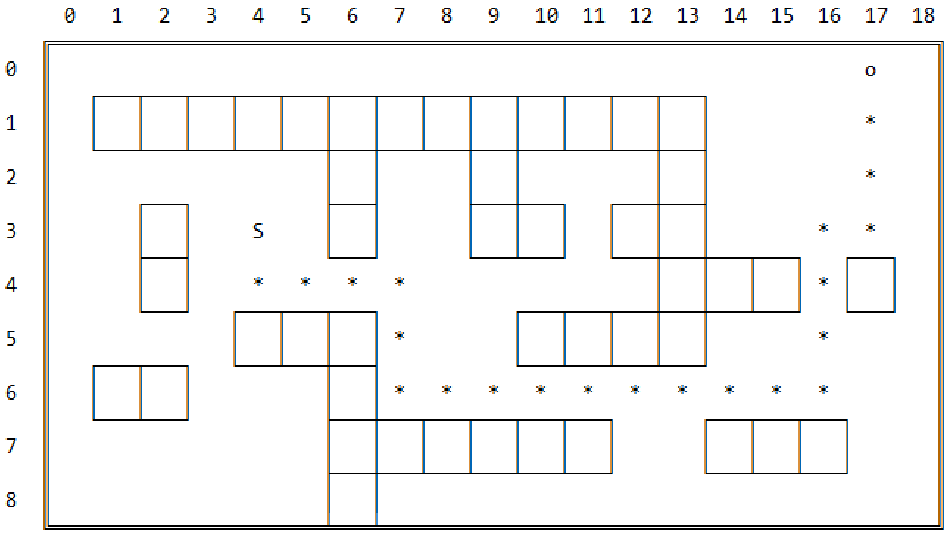
1.Complete the Java program that will read the given text file (maze.txt) and prints the output text files (dfs.txt and bfs.txt). You will only implement the following "if-else" sections (inside BFS_Solver and DFS_Solver classes under solve() methods):

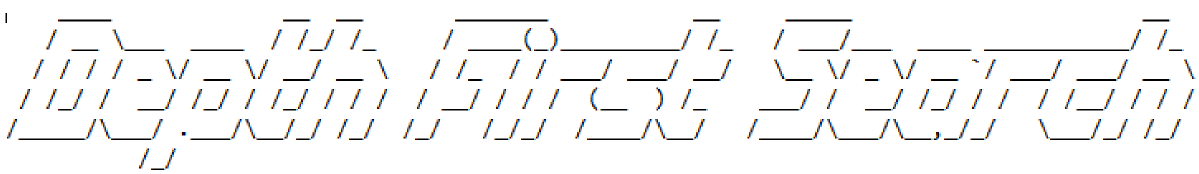
```
//checking if we have found the solution
if(currState.getLine() == this.maze.getEnd().getLine() && currState.getCol() == this.maze.getEnd().getCol())
{
        //You should create a new Node<T> for this maze to be able to assign a new father to it
        //Set current as father (parent) for all next states
        //push the goal (end) so you can reach starting point by using fathers (parents)
        endfound = true;
}
else
{
        LinkedList<Node<Maze>> nexts = this.getNextSquares(); //Get next possible states

        //now the current one was visited (that you want to explore its next states)
        //So it should be pushed to other stack (which keeps track the visited ones)
        //do not fotget to check if it is already in
        //do not forget to set * for the visited one (current one) (is it gonna be the Node or Square?)


        //You may want to use an iterator to reach every possible next state one by one
        //or you may want to get size of the next possible states and reach them accordingly


        //Add next possible states to stack
        //(think about which stack it could be.)
        //Do not forget to add all next possible states (at most they are: w s e n)
        //-> to be able to have w s e n, you should iterate it from end to start and push it like that
        //Do not forget to set father (parent) node of the pushed ones before pushing (as current node)
        //Do not forget to increment nodesCounter


        //You may need typecasting for the Stack (before pushing) as can be observed from given codes
}
```

Sample Outputs:

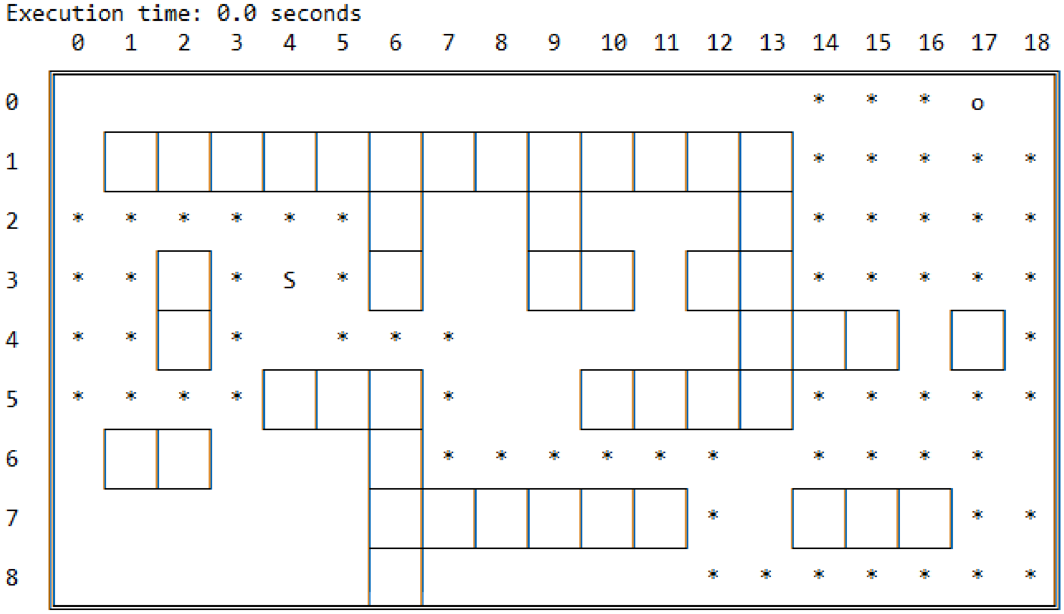


```
Path: [0, 17](0.0)(End) <- [1, 17](0.0) <- [2, 17](0.0) <- [3, 17](0.0) <- [3, 16](0.0) <- [4, 16](0.
Path length: 22
Number of nodes created: 167
Execution time: 0.0 seconds
```





```
Path: [0, 17](0.0)(End) <- [0, 16](0.0) <- [0, 15](0.0) <- [0, 14](0.0) <- [1, 14](0.0) <-
<- [3, 3](0.0) <- [3, 4](0.0)(Start)
Path length: 66
Number of nodes created: 151
Execution time: 0.0 seconds
```



You will send **ONLY** the codes of “if-else” sections in 2 .txt or .java files (1 for DFS and 1 for BFS).

BFS and DFS codes are not that different (once you figure out one, you can easily complete the other). You do not need to worry so much about other classes, just focusing on BFS or DFS_Solver class is going to be more than enough for you (this doesn’t mean that you should not look at them, look if you need). Read carefully the comments I left inside while loop. Even you are not very familiar with generics, do not panic, you can check other codes given to you (even in the same class), then figure out how to call-use-do it.

You should also prepare a simple report in a .pdf file format explaining how the “while(!endfound)” and “if(endfound)” sections work. It should only contain information about “how it works” and nothing more. No more than 300 words are allowed. You are free to use figures if it enriches your explanation (do not exceed 1 page). Also include your name and ID info to the report.

At the end of your report, rate the difficulty of this assignment on a scale of 1 to 10 (1 being the easiest homework ever, 10 being the hardest homework ever). Also, just leave an answer to the “Would you like your next assignment to be similar?” question.

This is the class diagram of the system: