A* LABORATORY PROJECT

In this project, you will develop an agent that is able to find its way in a maze full of obstacles. You must code a Java console application that creates the maze at random, applies the A* algorithm to find an optimal path, and prints the results to the console.

You must define a maze as a rectangular matrix of 60 rows and 80 columns. Each element of the matrix is a possible state of the agent.

A certain fraction of the matrix elements (say 30%) will contain an obstacle. The obstacles will be chosen at random. The initial and goal states for the agent will also be chosen at random. Therefore, for each execution of the Java application, they should be different. You must check that neither the initial nor the goal states are occupied matrix elements. In case that the initial or the goal states are occupied by obstacles, the program must output a message explaining the problem, and halt.

The agent must use the A* algorithm with a suitable heuristic function in order to find a sequence of actions (moves) that takes it from the initial state to the goal state. Then it must move according to the sequence to reach the goal. You should implement the A* algorithm by following the pseudocode provided in the slides of the search unit. You must design your solution, including the data structures that you think are most appropriate for this project.

When obstacles do not occupy the initial and the goal states, the output of the program must be a text of 60 rows and 80 columns, i.e., 4800 text characters total. Each character must be:

-An asterisk (*) if the state is an obstacle.

-A capital I letter (I) for the initial state.

-A capital G letter (G) for the goal state.

-A plus sign (+) if the state belongs to the optimal path found by the A* algorithm.

-A blank space, in all other cases.

If the goal state is not reachable from the initial state, then the program must output the 480 text characters without printing any optimal path, and after that, it must output a message explaining the problem.

A plain text file named output.txt must be generated which contains the full output of the program.

Optional task:

Analyze and discuss the mean length of the optimal path, and the number of times that the goal state is not reachable, as you change the fraction of obstacles in the matrix.

Tentative schedule:

Laboratory session 1: develop the code to generate, manage, and print out a maze.

Laboratory session 2: develop an implementation of the A* algorithm.

Laboratory session 3: present the final result to the class with the help of a slide show and do a practical demonstration of the software (maximum 10 minutes overall). All the materials (slide show, source code and output.txt file) must be submitted to the associated virtual campus task.