

Deadline: 8th September, 2024 midnight.

Consider the following grid:

[Total marks 60]

X		End
Start		X

You have to plan for a robot to go from start to end. The robot can only move right or up. Also, the robot cannot visit the grids marked as 'X'.

1. Model the puzzle as a class named Puzzle using the above state representation. Define methods that allow the puzzle to transition from state to state on a move. The move() method can take a single integer argument that can have values 0, 1, 2, 3, that represents moving the robot to the right, up, respectively. If a particular move is impossible then it should return the state None. Also, implement a method named play() that takes a single positive integer as argument, say k, and makes k moves starting with the current state of a puzzle object. Each move is chosen randomly from the set of possible moves in the current state at each stage. [10]
2. Design a problem graph to represent the above play. [10]
3. Implement a) depth first search, b) breadth first search, c) depth limited search and d) iterative deepening search. You can use default arguments in the generic graph method to pass necessary arguments to reorder OPEN as needed for each type of search. Use the four uninformed search algorithms in the previous question to search for a solution to the same puzzle from a start state (given by the user) to an end state. [10,10,10,10=40]