Artificial Intelligence-Lab

Lab 02:	Introduction to Search Algorithms in AI, uniformed search BFS & DFS

The objective of this is to learn the behavior of BFD and DFS for uniformed search.

Breadth First Search

Breadth-First Search is a "blind" algorithm. It's called "blind" because this algorithm doesn't care about the cost between vertices on the graph. The algorithm starts from a root node (which is the initial state of the problem) and explores all nodes at the present level prior to moving on to the nodes at the next level. If the algorithm finds a solution, returns it and stops the search, otherwise extends the node and continues the search process. Breadth-First Search is "complete", which means that the algorithm always returns a solution if exists. More specifically, the algorithm returns the solution that is closest to the root, so for problems that the transition from one node to its children nodes costs one, the BFS algorithm returns the best solution. In addition, in order to explore the nodes level by level, it uses a **queue data structure**, so new nodes are added at the end of the queue, and nodes are removed from the start of the queue.

Depth First Search

The Depth-first Search is a "blind" search algorithm that can be used to find a solution to problems that can be modelled as graphs.

It's called "blind" because this algorithm doesn't care about the cost between vertices on the graph. The algorithm starts from a root node and explores as far as possible along each branch before backtracking. if the algorithm finds a solution then it returns the solution and stops the search.

Task 1:

You are required to Implement the BFS and DFS on the following maze game.

A	1	1	1
0	1	1	0
0	0	0	0
0	1	1	0
0	0	1	G

- Take the 2D-array to store the above matrix
- Follow the 0 path to go from State A to State G
- 1 path is blocker you cannot pass through it

Approach:

- Start with a frontier that contains the initial state.
- Start with an empty explored set.
- Repeat:
    - If the frontier is empty, then no solution.
    - Remove a node from the frontier.
    - If node contains goal state, return the solution.
    - Add the node to the explored set.
    - Expand node, add resulting nodes to the frontier if they aren't already in the frontier or the explored set.

After applying DFS, BFS write the performance measure of both algorithms.

Submission Format:

File Naming	Roll_No_Lab _02.ipynb
Lab Report	Roll_No_Lab _02.pdf
Submission	On teams

Note: If you do not follow the submission guideline result in zero for the particular lab.