Introduction to Artificial Intelligence
Lab 2

Depth-first search and Breath-first search

In this laboratory we analyse the Depth-first search (DFS) and Breath-first search (BFS) algorithms compared to the A* search algorithm (Lab1). A report should be deposited on the moodle until 18 April 23:55

Part A.
Question 1.
Give the time complexity of DFS and BFS algorithms. Compare with A* and conclude.

Question 2.
What is the Memory Requirements for DFS, BFS and A*? Compare.

Question 3.
Make a comparison between DFS, BFS, Uniform Cost Search (UCS), A* and explain in which situations should be used one of them.

Part B.
Depth First Search (DFS)
DFS is a recursive algorithm that makes use of the concept of backtracking. It entails comprehensive searches of all nodes, either forward or backtracking if possible.
When you are traveling ahead and there are no more nodes along the current path, you retrace along the same path in order to discover new nodes to traverse. All nodes on the current path will be visited until all unvisited nodes have been explored, at which point the new path will be picked.

Stacks can be used to implement DFS's recursive nature.
The fundamental concept is as follows:

- Select a beginning node and stack all of its nearby nodes.
- To pick the next node to visit, pop a node from the stack and push all its surrounding nodes into a stack.
- This procedure should be repeated until the stack is empty. However, verify that the visited nodes are noted. This will prohibit you from returning to the same node several times. If you do not mark the visited nodes and return to the same node several times, you may wind yourself in an unending loop.

Question 1.
Writhe a Python functions corresponding to BFS algorithm by verifying the visited nodes.

Breadth First Search (BFS)
There are several methods for traversing graphs. BFS is the most often employed technique.

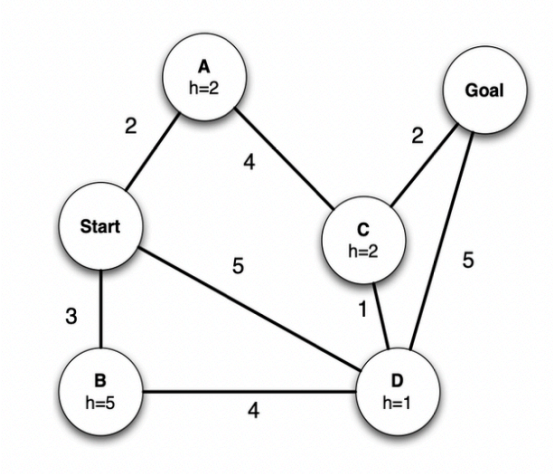
BFS is a graph traversal technique in which you should begin at a selected node (source or beginning node) and travel the graph layer by layer, thus discovering neighbour nodes (nodes which are directly connected to source node). After that, you must go to the next-level neighbour nodes.

As implied by the term BFS, you must traverse the graph breadthwise as follows:

- To begin, go horizontally and visit all nodes in the current layer.
- Proceed to the subsequent layer

Question 2.
Writhe a Python function corresponding to the DFS algorithm.

Part C.
Question 1.
Apply the both BFS and DFS algorithms ont the following graph :



Question 2.
Return the expanded nodes and the final result for both algorithms

Question 3.
Compare with the results obtained by A* in the Lab1.