Assignment No.1

Name - Kudharon Sumit Dattatraya
Roll No - 20U401
Class - TE4
subject -

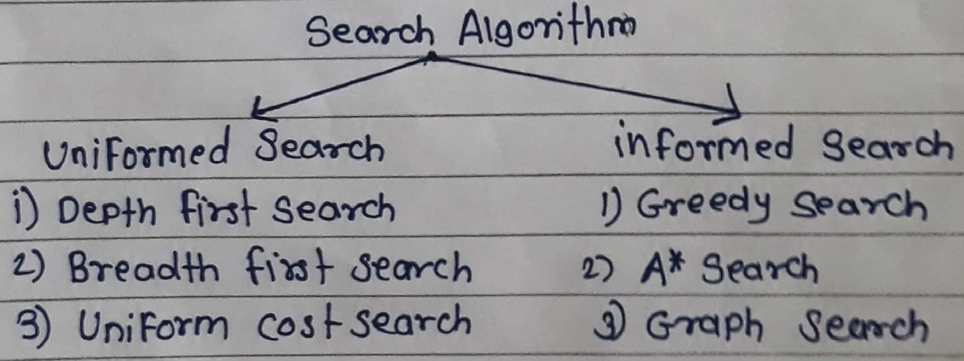
Problem Statement -

Implement depth first Search and breadth first Search algorithm. Use an undirected graph & develop a recursive algorithm for searching all the Vertices of a graph or tree data Structure.

Theory -

* Algorithm in AI.

- AI is the study of building agents that act rotationally.
- Most of the time these agents perform some kind of search algorithm in the background in order to achieve their tasks.
- A search Problem consist of State space start space goal test.
- A state space - Set of all possible creates where you can be.
- A goal test - A function that looks at the current state returns whether or not. It is the goal state.
- A start ~~State~~ state - The state from where the search begins.
- The solution to a search problem is a sequences of actions caused plan that transforms the start state to goal state.



Search Algorithm

Uniformed Search
1) Depth first Search
2) Breadth first search
3) Uniform cost search

informed Search
1) Greedy Search
2) A* Search
3) Graph Search

* what is informed & uniformed search algorithm in AI.

Informed Search:

Informed Search Algorithm have information on the goal State which helps in more efficient searching.

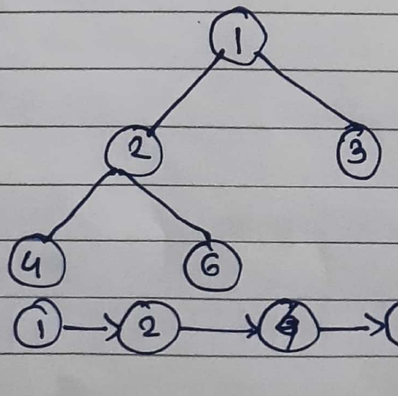
Uniformed Search:

Uniformed Search alogrithm have no additional information on the goal node other than the one provided in the Problem definition.

* Depth First Search (DFS).

- It is a recursive algorithm.
- It uses idea of back tracking.
- It involves exhaustive searches of all the nodes by going ahead if possible else by back tracking.
- Here, the word backtracking means that when you are moving forward & there are no more nodes along the current path.
- Then you move backword on the same path to find codes to traverse.
- All the nodes will be visited on the current path till the unvisited nodes have been traversed after which the next path will be selected.
- The recursive nature of DFS can be implemented using stack.

Example.



Solution - ① → ② → ④ → ⑤ → ③

Algorithm -

① set status = 1 i.e. ready state for each node G.
② Push the starting node (A) on the Stack & set its Status = 2 i.e. waiting state.
③ Repeat steps 4 & 5 Until the stack is empty.
④ POP the top node (N) process it & Set its Status = 3 i.e. procerced state.
⑤ Push on the stack all the neighbours of (N) that are in the ready state whose status = 1, set it as status = 2 & this will end the loop.
⑥ Stop.

Advantages -

① It requires very little memory as it only needs to store a stack of the nodes on the path from root node to current node.
② It takes less time to reach the goal node than BFS if it traverses in the eight path.

Disadvantages -

① There is the Probability that many states keep reoccurring & there is no gurantee of finding the solution.
② The DFS algorithm goes for deep down searchin & sometimes is many go into infinite loop.

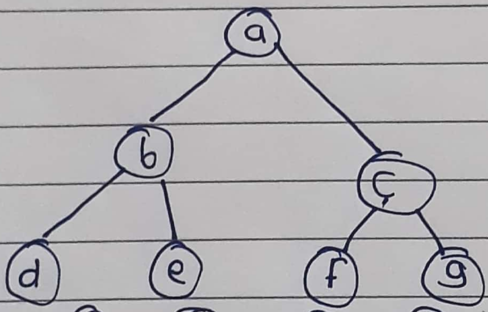
Applications -

① For finding the path.
② To test of graph is biportile.
③ Finding the strongly connected components of graph.
④ For detecting cycles in graph.

*Breadth First Search (BFS)-

- BFS is an algorithm for traversing or searching tree or graph data structures.
- It starts at the tree root & explore all of the neighbours at the present depth prior to moving on the nodes at the next depth level.
- The algorithm starts with an initial node & then Proceeds to explore all adjacent nodes in a breadth first Fashion.
- As opposed to DFS which goes down a particular branch till all nodes in the branch are visited.
- It means it traverses the graph level wise & not moving down a level till all the nodes In that level are visited & marked.

Example.



Solution - (a)→(b)→(c)→(d)→(e)→(F)→(g)

Algorithm -

① Assign (a) as the root node & insert it into the queue.
② Extract (a) from the queue & insert child nodes of (a)
③ Print node(a)
④ The queue is not empty & base node (b) & (c).
⑤ Since (b) is the first node in the queue extract it.
⑥ Insert the child node of (b).
⑦ Repeat above steps ~~till~~ untill the queue gets empty.
⑧ Exit.

Advantages-

① It will provide a solution if any exists.

② If there is more than one solution for agiven Problem then BFS will provide the optimal solution.

Disadvantages.

① It requires a lot of memory since each level of tree must be saved into memory.

② BFS needs a lot of time if the solution is far away from root node.

Application-

① To build index try search index

② For GPS Navigation.

③ Path finding algorithm.

④ In ford-falkesson algorithm to find maximum flow in a network.

⑤ Cycle desection in an undirected graph.

* compare DFS and BFS.

	BFS	DFS
①	BFS stands for Breadth first search.	DFS stands for Depth first search.
②	BFS uses queue data structure for finding the shortest path	DFS uses stack data structure.
③	BFS is more ~~schedular~~ suitable for searching vertices which are closer to the given source.	DFS is more suitable when there are solutions away from source.
④	BFS considers all neighbours first & therefore suitable.	DFS is more suitable for game or puzzle problems we make a decision then explore all paths through this decision.

Conclusion -

Learned & implemented Depth First Search & Breadth First Search Algorithm.