Depth first search

Introduction

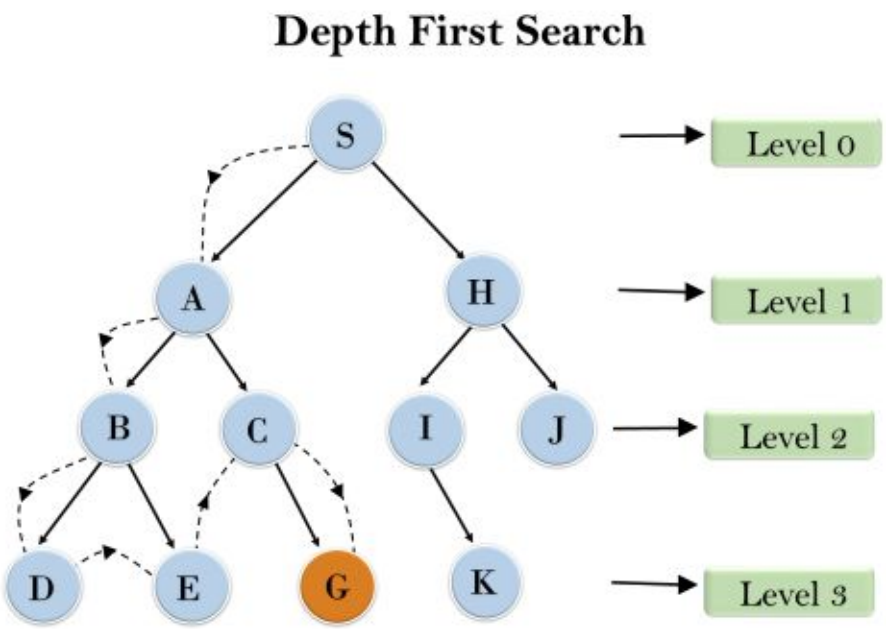
- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation.
- The process of the DFS algorithm is similar to the BFS algorithm.

- **Advantage:**
- DFS requires very less memory as it only needs to store a stack of the nodes on the path from root node to the current node.
- It takes less time to reach to the goal node than BFS algorithm (if it traverses in the right path).
- **Disadvantage:**
- There is the possibility that many states keep re-occurring, and there is no guarantee of finding the solution.
- DFS algorithm goes for deep down searching and sometime it may go to the infinite loop.

Example:

- In the below search tree, we have shown the flow of depth-first search, and it will follow the order as:
- Root node--->Left node ----> right node.

Example:

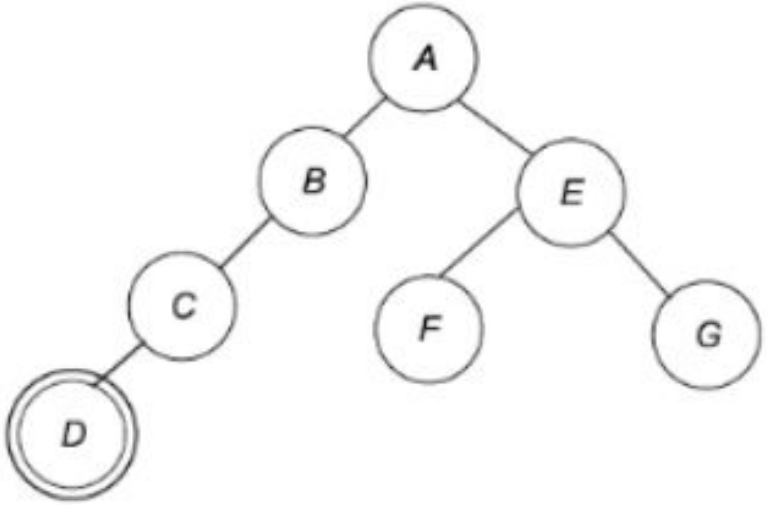


It will start searching from root node S, and traverse A, then B, then D and E, after traversing E, it will backtrack the tree as E has no other successor and still goal node is not found. After backtracking it will traverse node C and then G, and here it will terminate as it found goal node.

Example :

	Stack	
Step 1:	$A \leftarrow$ top $B \leftarrow$ top E	Pop A, is it goal? No, push its children on the stack. A's successor is pushed.
Step 2:	$B \leftarrow$ top E $C \leftarrow$ top E	Pop B, is it goal? No, push its children on the stack. B's successor is pushed.
Step 3:	$C \leftarrow$ top $D \leftarrow$ top E	Pop C, is it goal? No, push its children on the stack. C's successor is pushed.
Step 4:	$D \leftarrow$ top E	Pop D, is it goal?? YES!

Pop all the contents. Path found!



Depth limited search

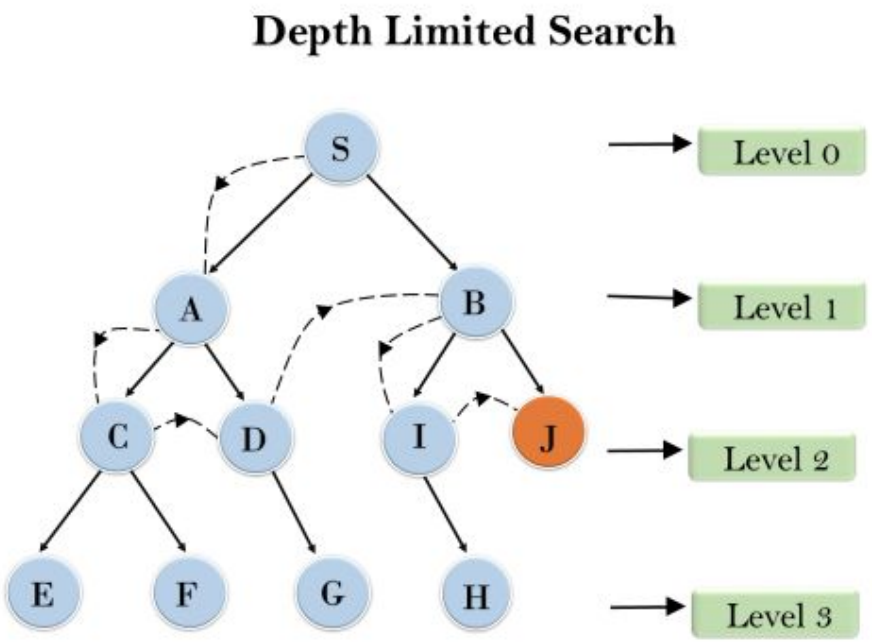
- A depth-limited search algorithm is similar to depth-first search with a predetermined limit. Depth-limited search can solve the drawback of the infinite path in the Depth-first search. In this algorithm, the node at the depth limit will treat as it has no successor nodes further.
- Depth-limited search can be terminated with two Conditions of failure:
- Standard failure value: It indicates that problem does not have any solution.
- Cutoff failure value: It defines no solution for the problem within a given depth limit

- **Advantages:**
- Depth-limited search is Memory efficient.
- **Disadvantages:**
- Depth-limited search also has a disadvantage of incompleteness.
- It may not be optimal if the problem has more than one solution.

The following algorithm summarises the working of depth limited search:

1. Set the depth limit to the maxdepth to search.
2. Initial node = current node
   If initial node = goal, return
3. If depth(initial node) > depth limit
   return
   else
   Expand(initial node)
   Save(successors) using stack
   Go to step 2

Example:



Completeness: DLS search algorithm is complete if the solution is above the depth-limit.

Time Complexity: $O(b^{\ell})$.

Space Complexity: $O(b \times \ell)$.

Optimal: Depth-limited search can be viewed as a special case of DFS, and it is also not optimal even if $\ell > d$.