Breadth First Search

- Breadth-first search is the most common search strategy for traversing a tree or graph.
- This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.
- BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- The breadth-first search algorithm is an example of a general-graph search algorithm.
- Breadth-first search implemented using FIFO queue data structure.

Breadth First Search

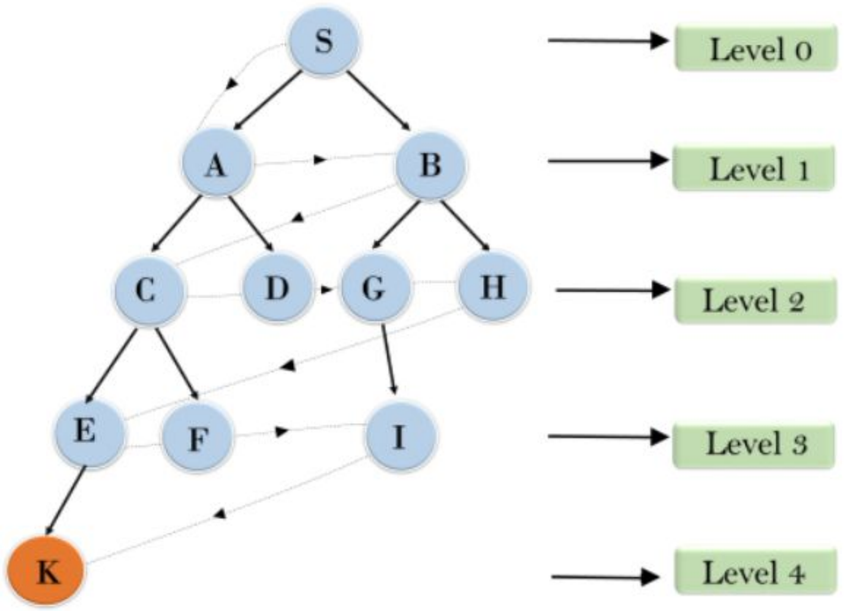
Advantages:

- BFS will provide a solution if any solution exists.
- If there are more than one solutions for a given problem, then BFS will provide the minimal solution which requires the least number of steps.

Disadvantages:

- It requires lots of memory since each level of the tree must be saved into memory to expand the next level.
- BFS needs lots of time if the solution is far away from the root node.

Breadth First Search



S---> A--->B---->C--->D---->G--->H--->E---->F---->I---->K

- The above figure is an example of a BFS Algorithm.
- It starts from the root node A and then traverses node B.
- Till this step, it is the same as DFS.
- But here, instead of expanding the children of B as in the case of DFS, we expand the other child of A, i.e., node C because of BFS, and then move to the next level and traverse from D to G and then from H to K in this typical example.
- To traverse here, we have only taken into consideration the lexicographical order.
- This is how the BFS Algorithm is implemented.

Breadth First Search

Time Complexity: Time Complexity of BFS algorithm can be obtained by the number of nodes traversed in BFS until the shallowest Node. Where the d= depth of shallowest solution and b is a node at every state.

$T(b) = 1+b^2+b^3+.......+ b^d = O(b^d)$

Space Complexity: Space complexity of BFS algorithm is given by $O(b^d)$.

Completeness: BFS is complete, which means if the shallowest goal node is at some finite depth, then BFS will find a solution.

Optimality: BFS is optimal

Depth First Search

- 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.

Depth First Search

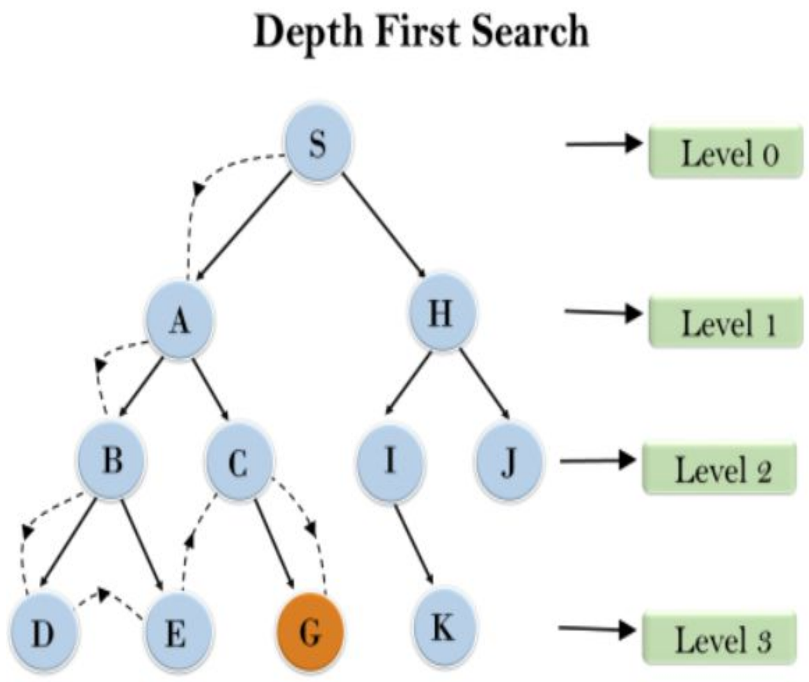
Advantage:

- 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 reoccurring, 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.

Depth First Search



- 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.**
- 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.

Depth First Search

Completeness: DFS search algorithm is complete within finite state space as it will expand every node within a limited search tree.

Time Complexity: $O(b^m)$

Space Complexity: DFS algorithm needs to store only single path from the root node, hence space complexity of DFS is equivalent to the size of the fringe set, which is **$O(bm)$**.

Optimal: DFS search algorithm is non-optimal, as it may generate a large number of steps or high cost to reach to the goal node.

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.

Depth Limited Search

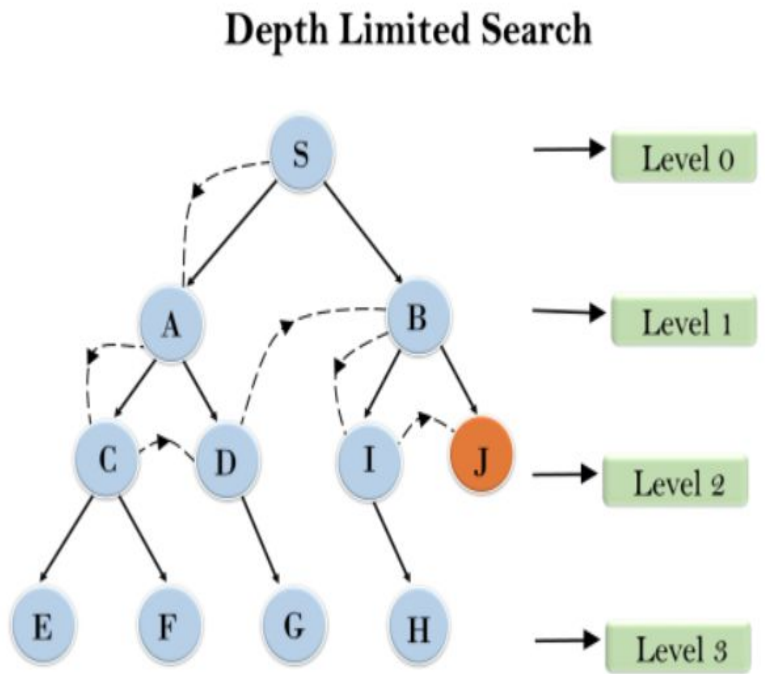
Advantages:

- 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.

Depth Limited Search



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

Time Complexity: Time complexity of DLS algorithm is **O(b$^{\ell}$)**.

Space Complexity: Space complexity of DLS algorithm is O(**b×ℓ)**.

Optimal: it is not optimal