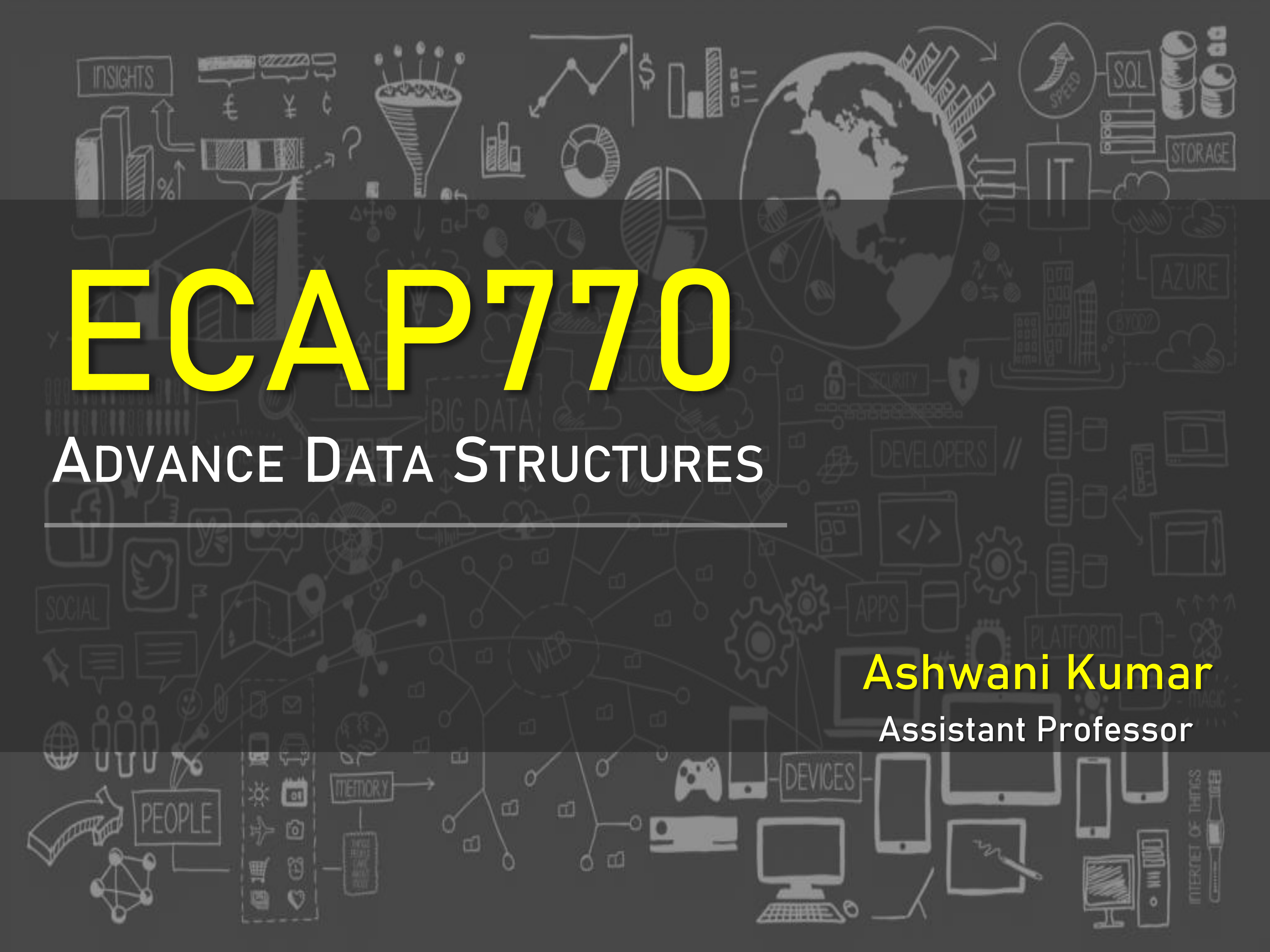


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand Depth first search

Graph traversals

- In Graph traversal visiting every vertex and edge exactly once in a well-defined order.
- In graph algorithms, you must ensure that each vertex of the graph is visited exactly once.
- The order in which the vertices are visited may depend upon the algorithm or type of problem going to solve.

Graph Traversal Algorithm

- Breadth First Search (BFS)
- Depth First Search (DFS)

Depth First Search

- DFS traversal is a recursive algorithm for searching all the vertices/ nodes of a graph or tree using stack data structure.
- In Depth First Search (DFS) algorithm traverses a graph in a depth ward motion.
- The DFS algorithm use the concept of backtracking.

Steps for DFS

- Step 1 – Visit the adjacent unvisited vertex. Mark it as visited. Display it. Push it in a stack.
- Step 2 – If no adjacent vertex is found, pop up a vertex from the stack. (It will pop up all the vertices from the stack, which do not have adjacent vertices.)
- Step 3 – Repeat Rule 1 and Rule 2 until the stack is empty.

Depth First Search

- For using DFS algorithm user should know about data structure Stack (*Last In First Out*) and its relevant operations like Push and Pop.

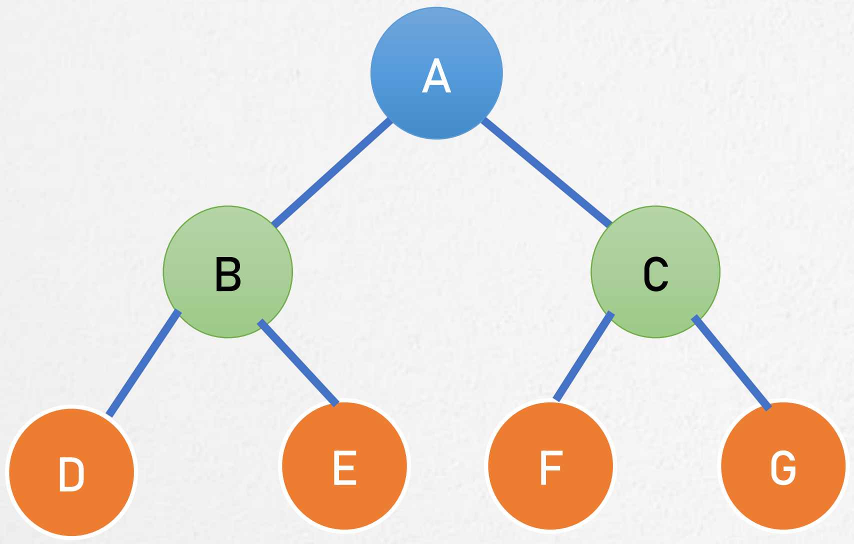
Algorithm: Depth First Search

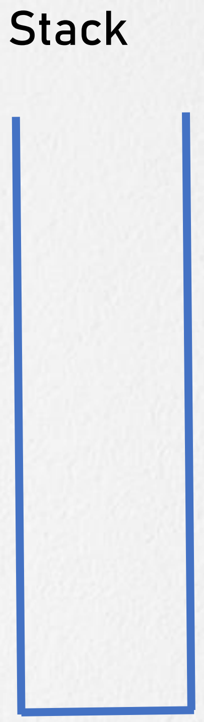
```
Step 1: SET STATUS = 1 (ready state) for each node
        in G
Step 2: Push the starting node A on the stack and
        set its STATUS = 2 (waiting state)
Step 3: Repeat Steps 4 and 5 until STACK is empty
```

Algorithm: Depth First Search

```
Step 4: Pop the top node N. Process it and set its
        STATUS = 3 (processed state)
Step 5: Push on the stack all the neighbours of N that
        are in the ready state (whose STATUS = 1) and
        set their
        STATUS = 2 (waiting state)
        [END OF LOOP]
Step 6: EXIT
```

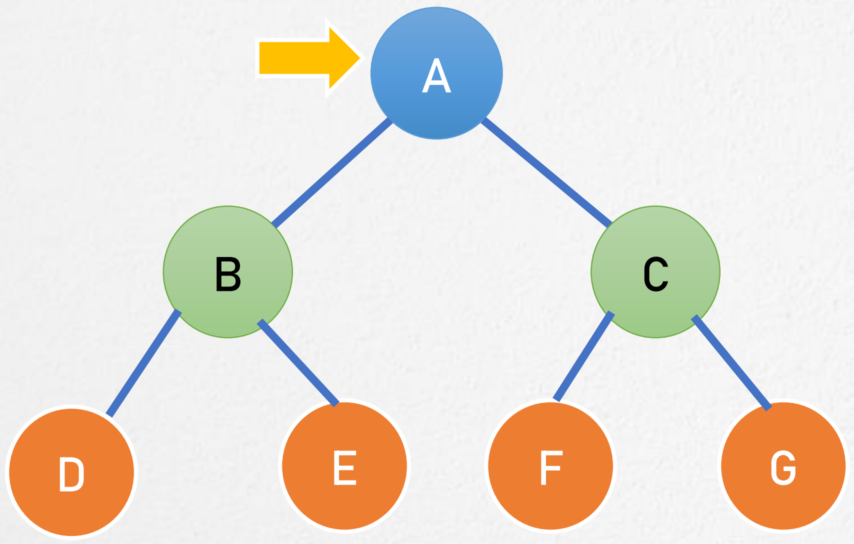
Example: Depth First Search

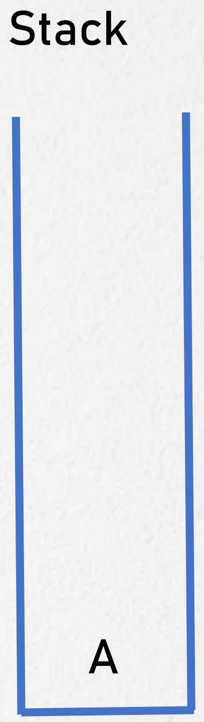




Output / visited :

Example: Depth First Search

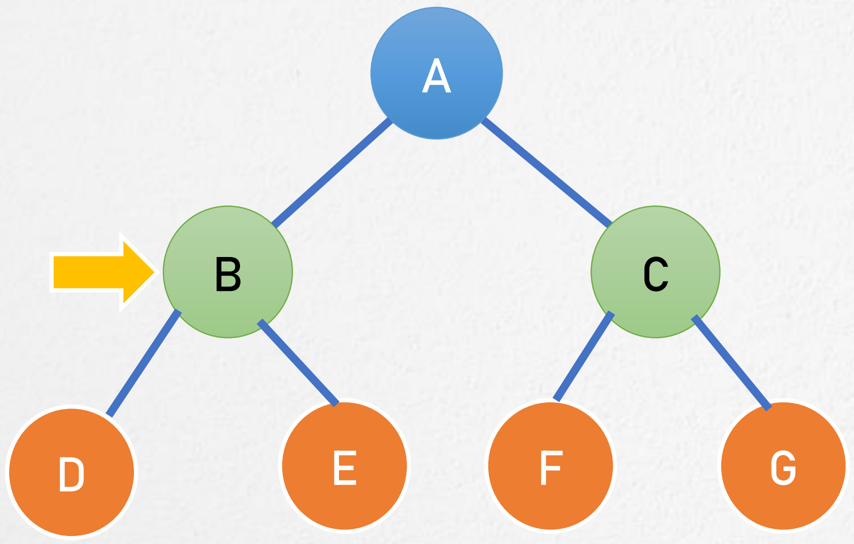


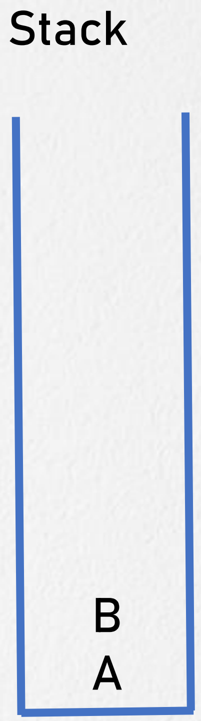


Output / visited : A

Fig: (A)

Example: Depth First Search

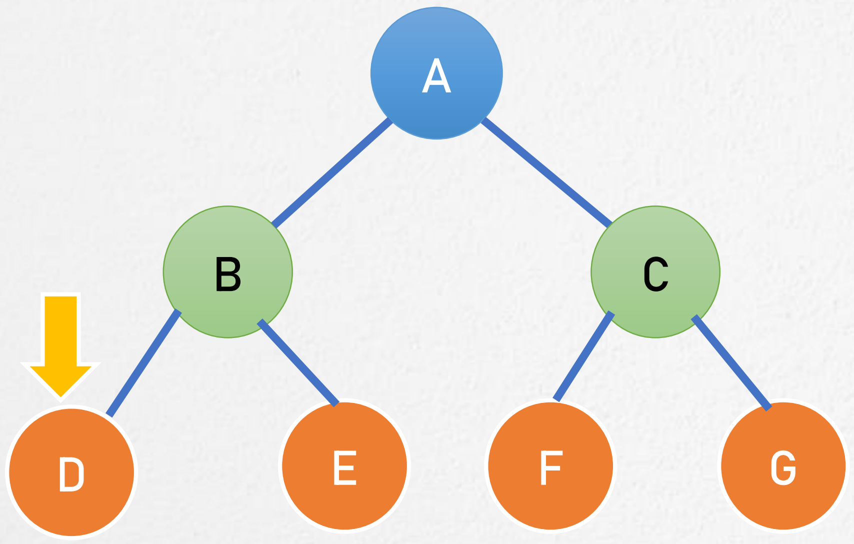


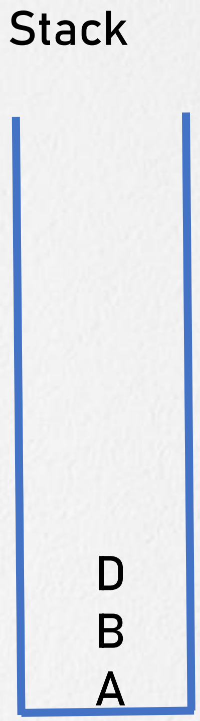


Output / visited : A B

Fig: (B)

Example: Depth First Search

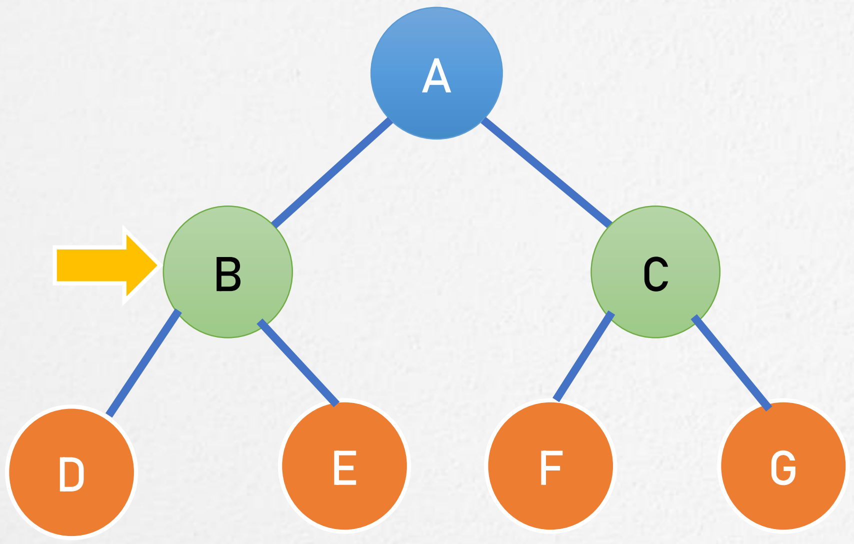


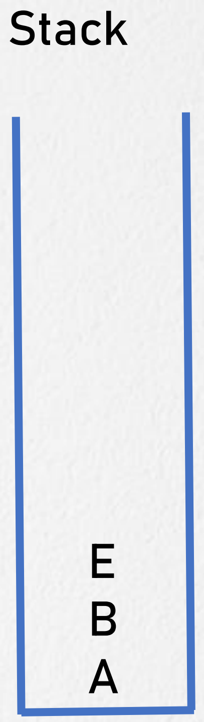


Output / visited : A B D

Fig: (C)

Example: Depth First Search

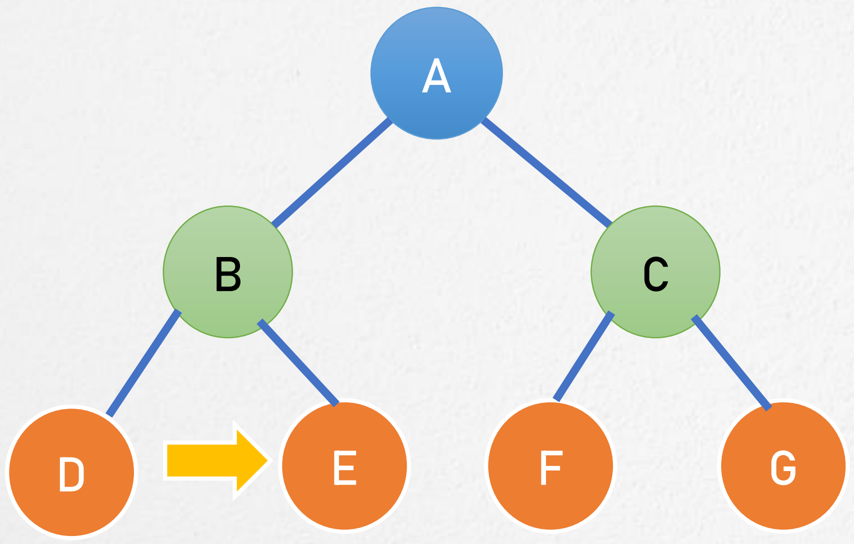




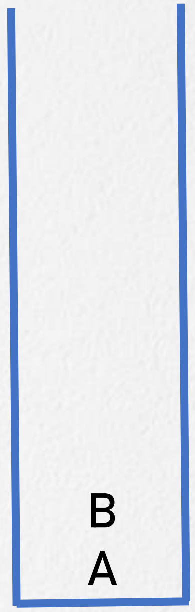
Output / visited : A B D E

Fig: (D)

Example: Depth First Search



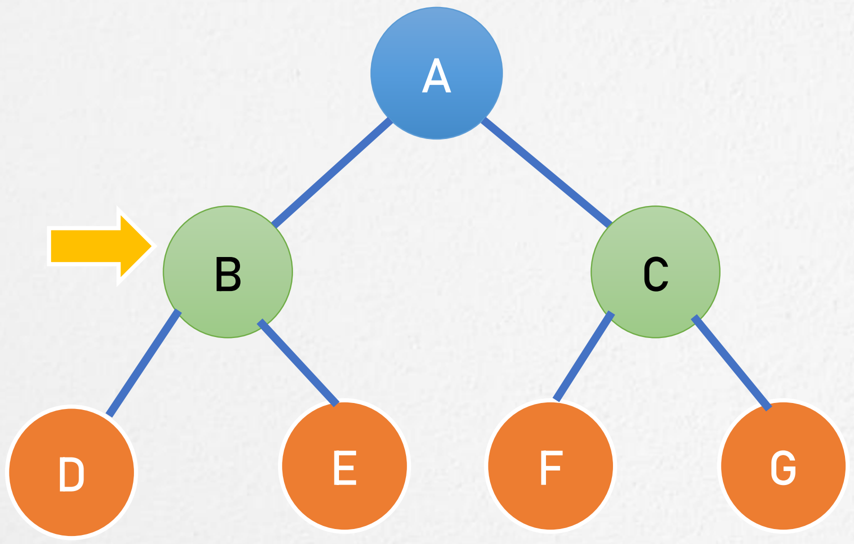
Stack

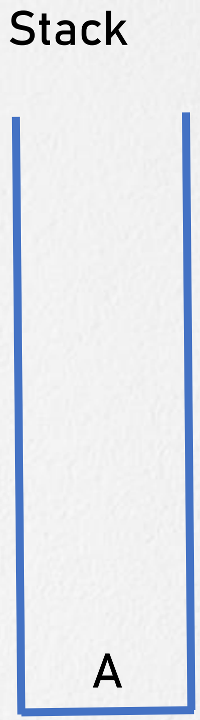


Output / visited : A B D E

Fig: (E)

Example: Depth First Search

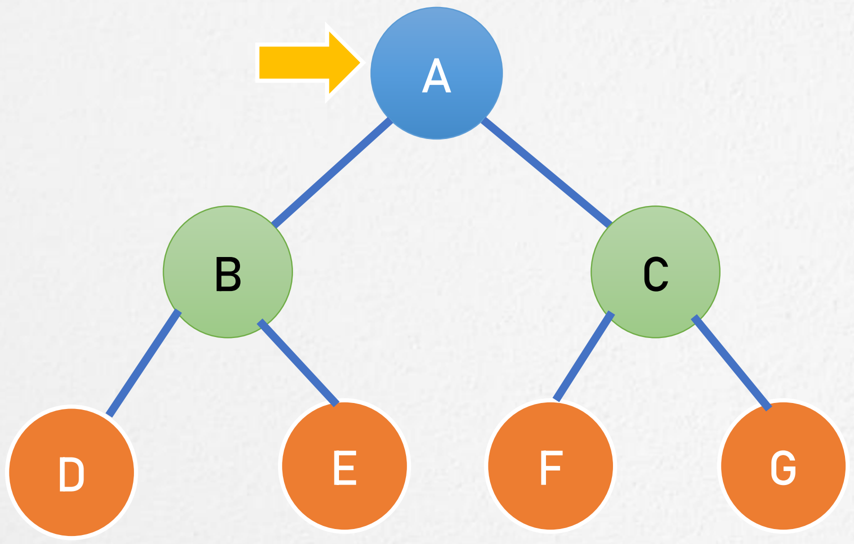


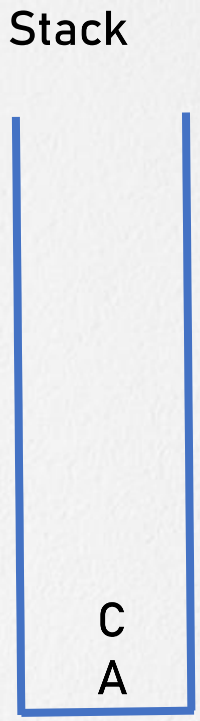


Output / visited : A B D E

Fig: (F)

Example: Depth First Search

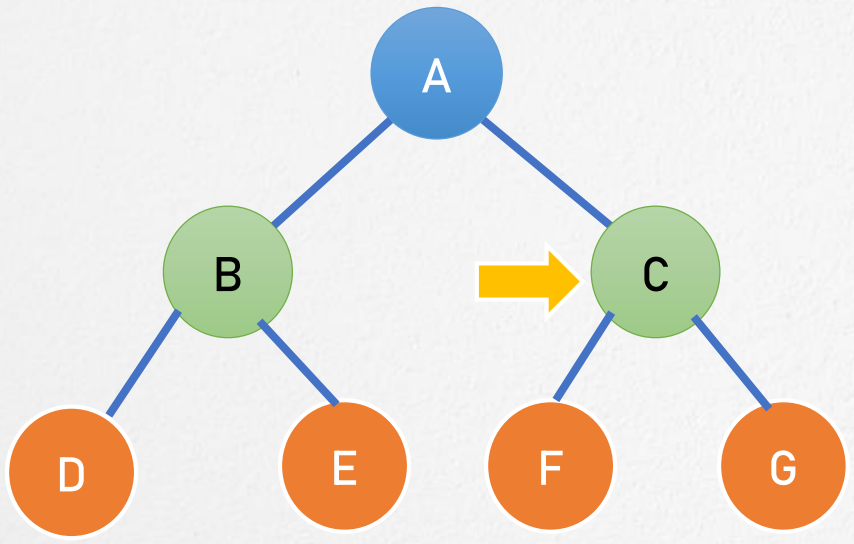


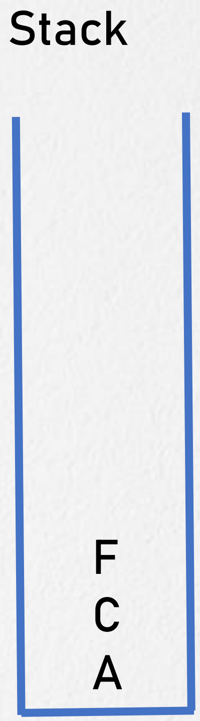


Output / visited : A B D E C

Fig: (G)

Example: Depth First Search

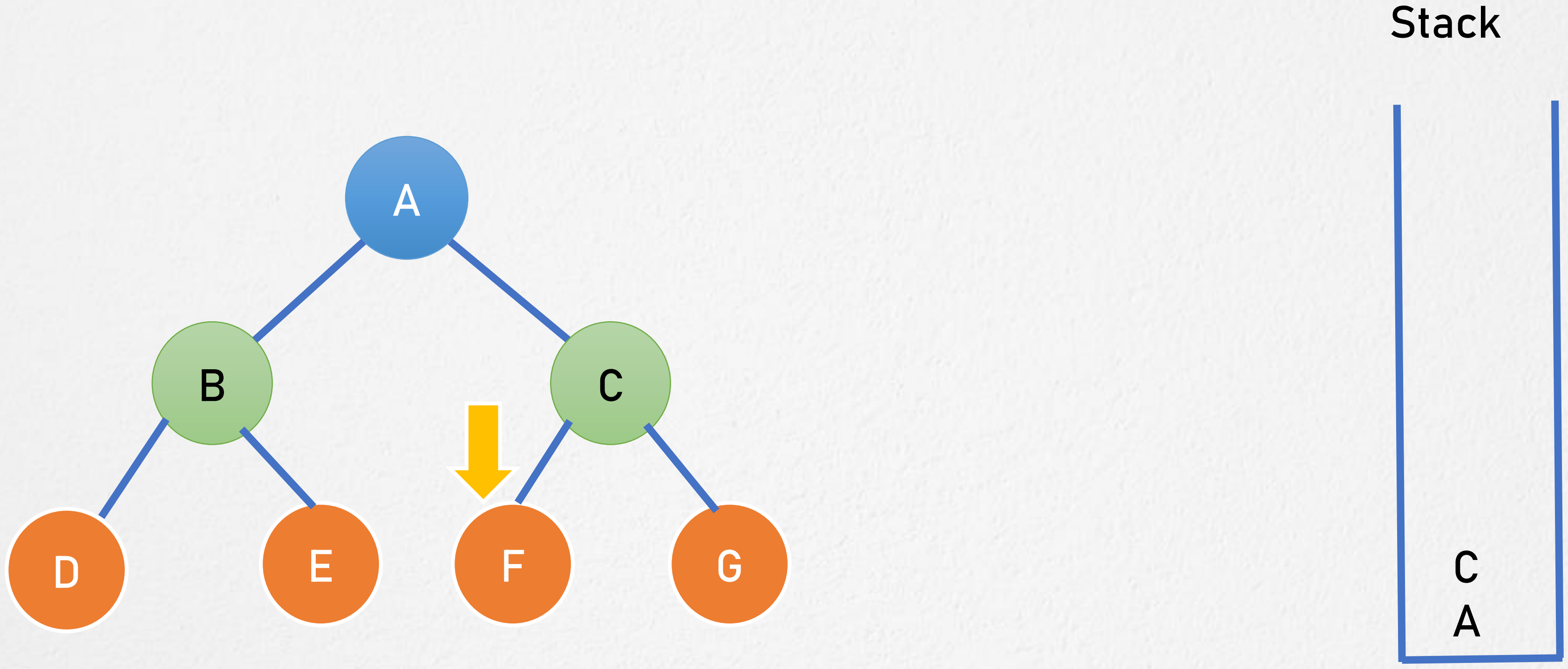




Output / visited : A B D E C F

Fig: (H)

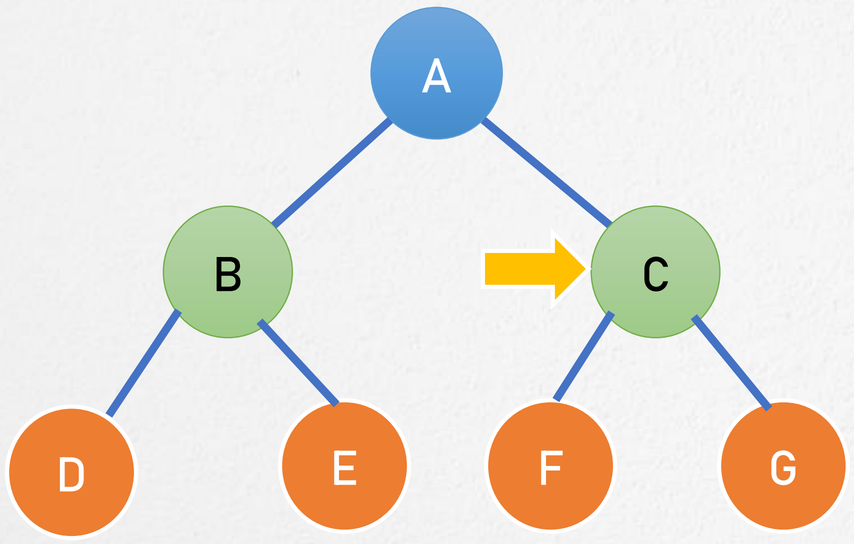
Example: Depth First Search

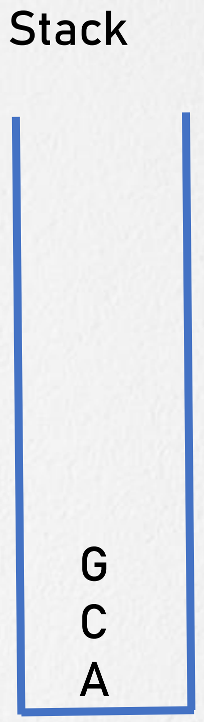


Output / visited : A B D E C F

Fig: (I)

Example: Depth First Search

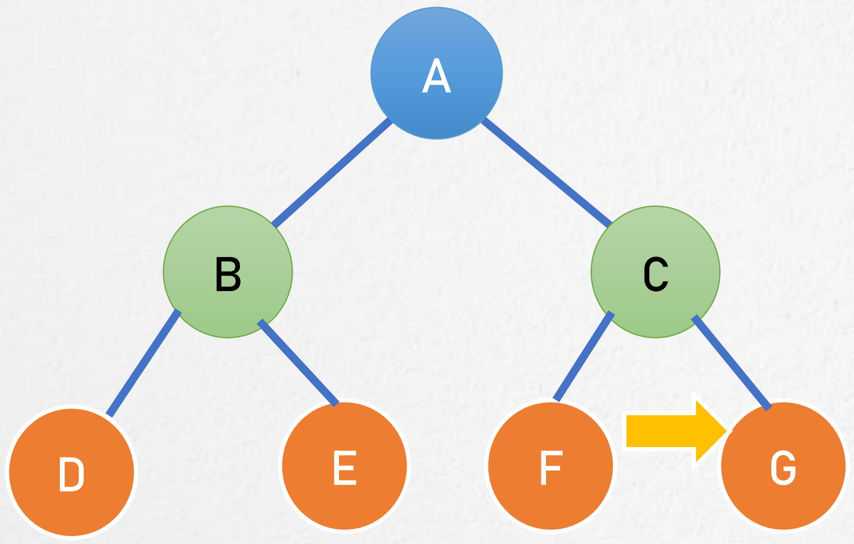


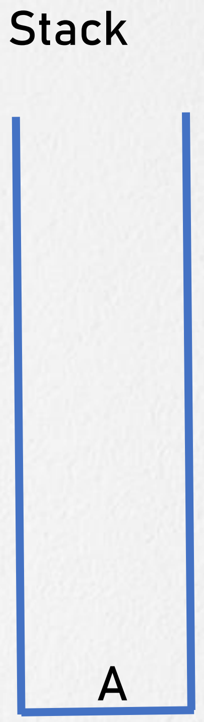


Output / visited : A B D E C F G

Fig: (J)

Example: Depth First Search

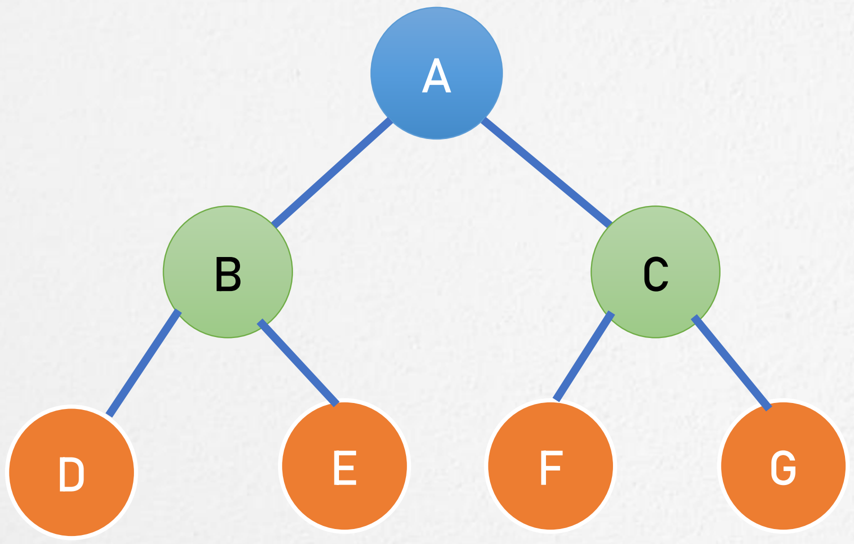




Output / visited : A B D E C F G

Fig: (K)

Example: Depth First Search



Empty Stack

Output / visited : A B D E C F G

Fig: (L)

Algorithm Complexity

- Time complexity: O(V + E), where V is the number of vertices and E is the number of edges in the graph.
- Space Complexity: O(V).

DFS Applications

- Mapping Routes and Network Analysis.
- Path Finding.
- Cycle detection in graphs.
- Topological Sorting.
- Solving puzzle.

That’s all for now…