

Tree Traversal Theory

Tree is a variant of graph where there exists only one way from root to any other node. Means we can apply graph traversal **Depth First Search(DFS)** and **Breadth First Search(BFS)** to tree.

DFS traversal classification is based on the position of ROOT NODE in traversal order :

DFS traversal possibilities for tree:

- 1. NLR 4. NRL
- 2. LNR 5. RNL
- 3. LRN 6. RLN

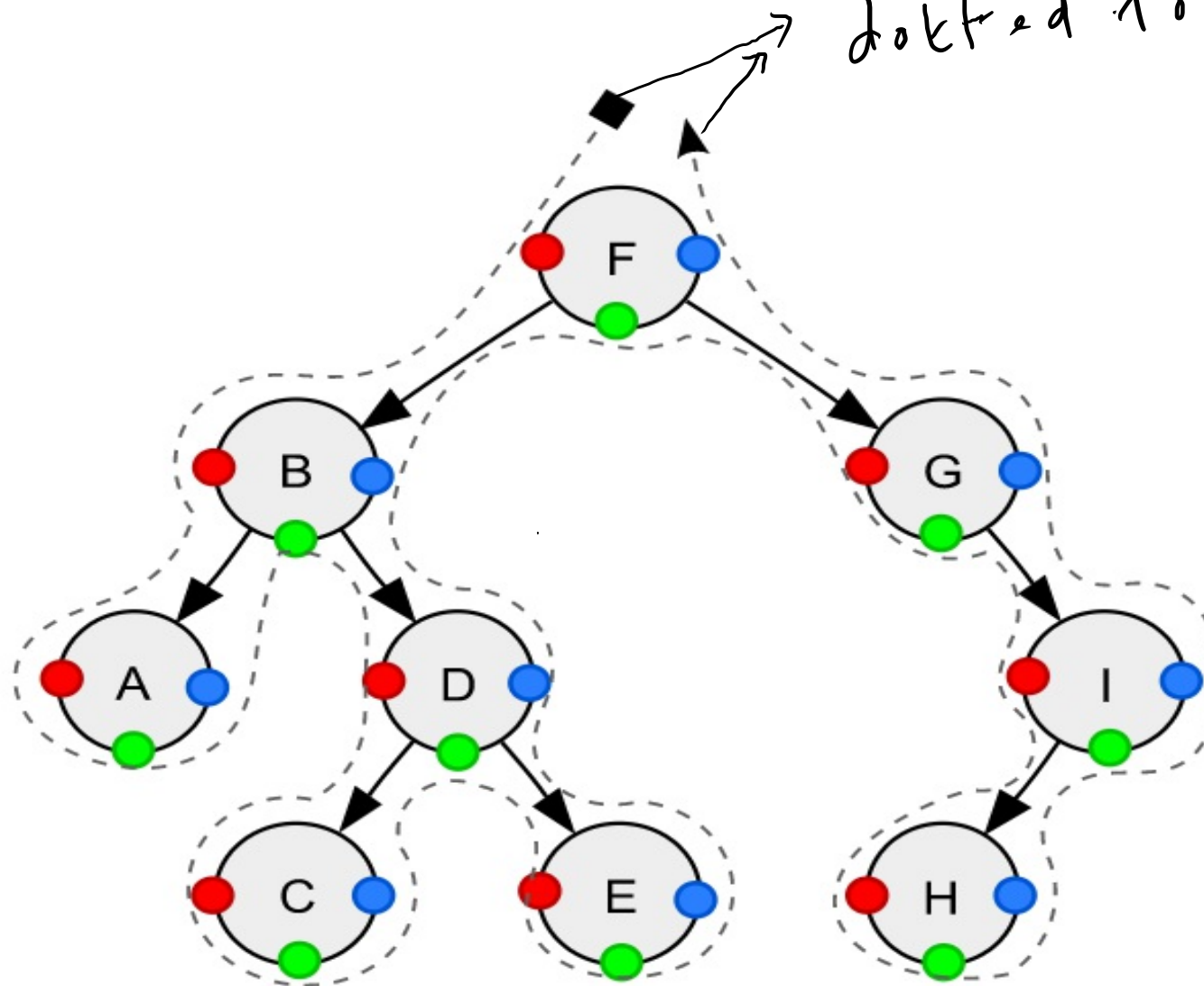
Following 3 unique DFS Traversal possibilities are available :

- 1. Preorder(**NLR**)
- 2. InOrder(**LNR**)
- 3. PostOrder(**LRN**)

BFS based tree traversal : is known as Level Order traversal.

Depth-first traversal (dotted path) of a binary tree:

- *Pre-order (node access at position red ●):*
F, B, A, D, C, E, G, I, H;
- *In-order (node access at position green ●):*
A, B, C, D, E, F, G, H, I;
- *Post-order (node access at position blue ●):*
A, C, E, D, B, H, I, G, F.



The trace of a traversal is called a ***sequentialisation*** of the tree.

No one sequentialisation according to pre-, in- or post-order describes the underlying tree uniquely.

Given a tree with distinct elements, either pre-order or post-order paired with in-order is sufficient to describe the tree uniquely.

However, pre-order with post-order leaves some ambiguity in the tree structure.

Properties of Pre-order(NLR) Traversal:

The pre-order traversal is a topologically sorted one while following the DFS path, because a parent node is processed before any of its child nodes is done.

Topological Sort :

In computer science, a topological sort or topological ordering of a directed graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v , u comes before v in the ordering.

Properties of In-order(LNR) Traversal:

InOrder(LNR) : Inorder traversal for BST always retrieves the keys in ascending sorted order.