DSA SERIES

- Learn Coding

Topic to be Covered today

Binary Tree Traversal

LETS START TODAY'S LECTURE

Lecture - 49

Traversing

Traversal = the process of visiting each node in a specific order.

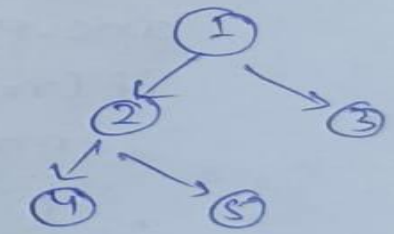
There are 2 main categories :-

i) Depth First Traversal (DFS)
   ↳ go deep before moving sideways.
   - Inorder
   - Preorder
   - PostOrder

ii) Breadth - First Traversal (BFS) → level by level.

Inorder (Left → Root → Right)

- Visit the left subtree
- Visit the root node
- visit the right subtree.

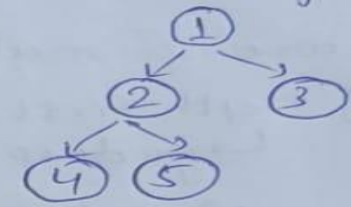


(4) → 2 → 5 → 1 → 3

Code.
```
void inOrder(root, ans){
    if(root==NULL) return;
    if(root->left){
        inOrder(root->left, ans);
    ans.push_back(root->val, );
    if(root->right){
        inOrder(root->right, ans);
        }
}
```

Preorder Traversal (Root → Left → Right)

- Visit the root node
- Visit the left subtree
- Visit the right subtree

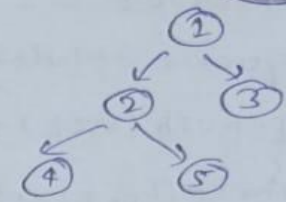


⇒ 1 → 2 → 4 → 5 → 3

Code.
```
void preorder(TreeNode* root, vector<int>& ans){
    if(root == NULL) return;
    ans.push_back(root->val);
    if(root->left){
        preorder(root->left, ans);
        }
    if(root->right){
        preorder(root->right, ans);
        }
}
```

* PostOrder Traversal (Left → Right → Root)

- Visit the left subtree
- Visit the right subtree
- Visit the root node.



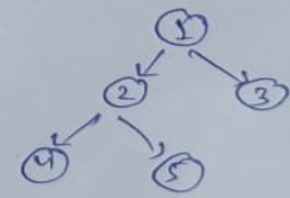
4 → 5 → 2 → 3 → 1

code.

```
void postOrder(Node * root, vector<int>& ans)
{
if (root == NULL) return;
if(root->left) {
postOrder(root->left, ans);
}
if (root->right){
   postOrder(root->right, ans);
  }
ans.push_back(root->val);
}
```

* Level Order Traversal (BFS)

- Visit nodes level by level, from left to right.



1 → 2 → 3 → 4 → 5

```
void levelOrder (Node * root) {
    if (root == NULL) return ~~NULL~~;
    queue <Node*> q;
    q.push (root);
   while (!q.empty()){
    Node * node = q.front();
     q.pop();
    cout << node->data << " ";
    if (node->left ) q.push (node->left);
    if (node->right) q.push ( node->right);
     }
}
```

Leetcode Question

Q. 94
Q. 145
Q. 144

Learn coding

THANK YOU