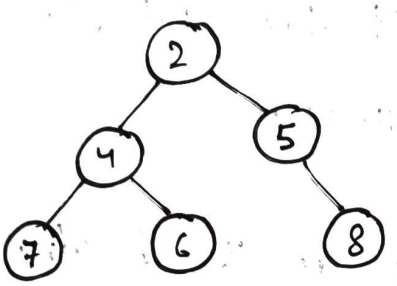
Binary Tree PreOrder Traversal (144)

Preorder traversal is visiting node, then left child, then right child.

Ex.



Preorder Traversal

= [2, 4, 7, 6, 5, 8].

Recursive solution is Trivial. Iterative solution is what counts.

Recursive.

```
struct TreeNode {
        int val;
        TreeNode * left;
        TreeNode * right;
    };

class Solution {
    public:
        vector <int> v;
        vector <int> preorder ( TreeNode* root) {
            if (root == NULL) {
                return v;
            }
            v.push_back ( root -> val);
            preorder (root -> left);
            preorder (root -> right);
            return v;
    }; }
```

For iterative solution, we will maintain a stack of TreeNode pointers which stores the right child of visited nodes. if there is a left child of that node.

Iterative

```
vector <int> v;
vector <int> preorderTraversal (TreeNode* root) {

    stack <TreeNode*> rightNodes;

    while (root != NULL) {
        v.push_back (root -> val);
        if (root -> left == NULL && root -> right == NULL) {
            if (rightNodes.empty()) break;
            else {
                root = rightNodes.top();
                rightNodes.pop();
            }
        }
        else if (root -> left != NULL && root -> right == NULL) {
            root = root -> left;
        }
        else if (root -> left == NULL && root -> right != NULL) {
            root = root -> right;
        }
        else {
            rightNodes.push (root -> right);
            root = root -> left;
        }
    return v; }
```