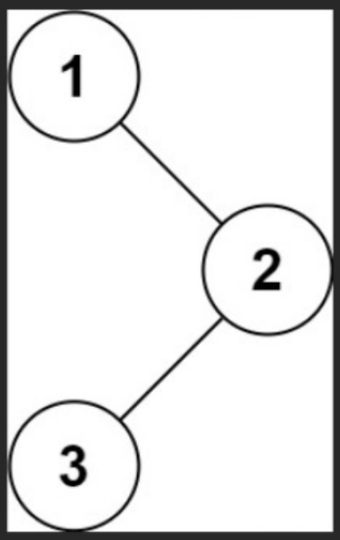
144. Binary Tree Preorder Traversal

Easy 7.4K 193

Companies

Given the `root` of a binary tree, return *the preorder traversal of its nodes' values.*

Example 1:




```
Input: root = [1,null,2,3]
Output: [1,2,3]
```

Example 2:

```
Input: root = []
Output: []
```

Example 3:

```
Input: root = [1]
Output: [1]
```

Constraints:

- The number of nodes in the tree is in the range `[0, 100]`.
- `-100 <= Node.val <= 100`

Follow up: Recursive solution is trivial, could you do it iteratively?

Approach 1: Recursive implementation

< val left right >

```
Preorder ( root )
{
    if (root is null) return

    Print val
    Preorder ( root → left )
    Preorder ( root → right )
}
```

T(n) : O(n)
S(n) : recursion stack space

Approach 2: Iterative implementation

→ Curr = root

```
while ( Curr is not null // Stack is not empty )
{
    if ( Curr is not null )
    {
        add val to ans
```

```
        Push Curr to stack
        Curr = Curr → left
    }
    else means we found a node without
    {   LST So now explore its right.
        temp = top of stack
        pop
        Curr = temp → right
    }
}
```

$T(n) : O(n)$
$S(n) : O(n)$