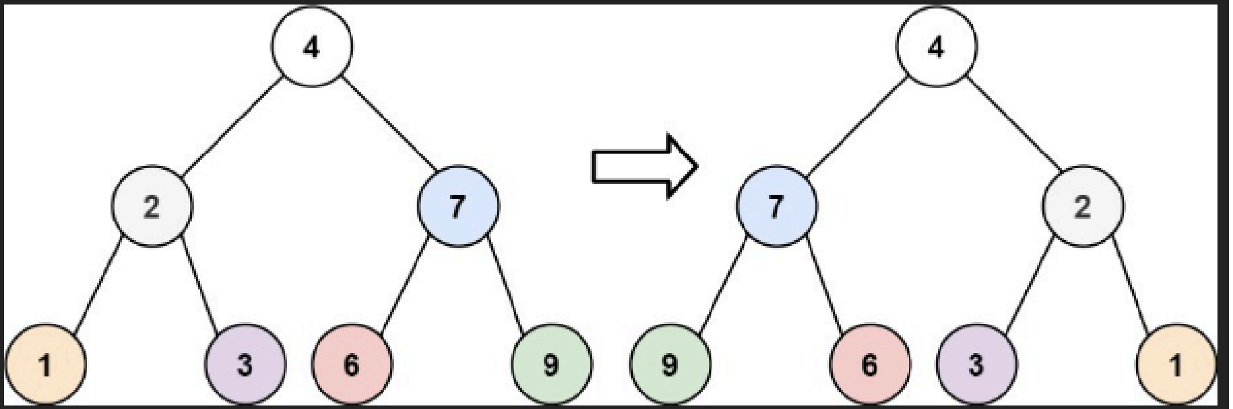
226. Invert Binary Tree

Problem Statement

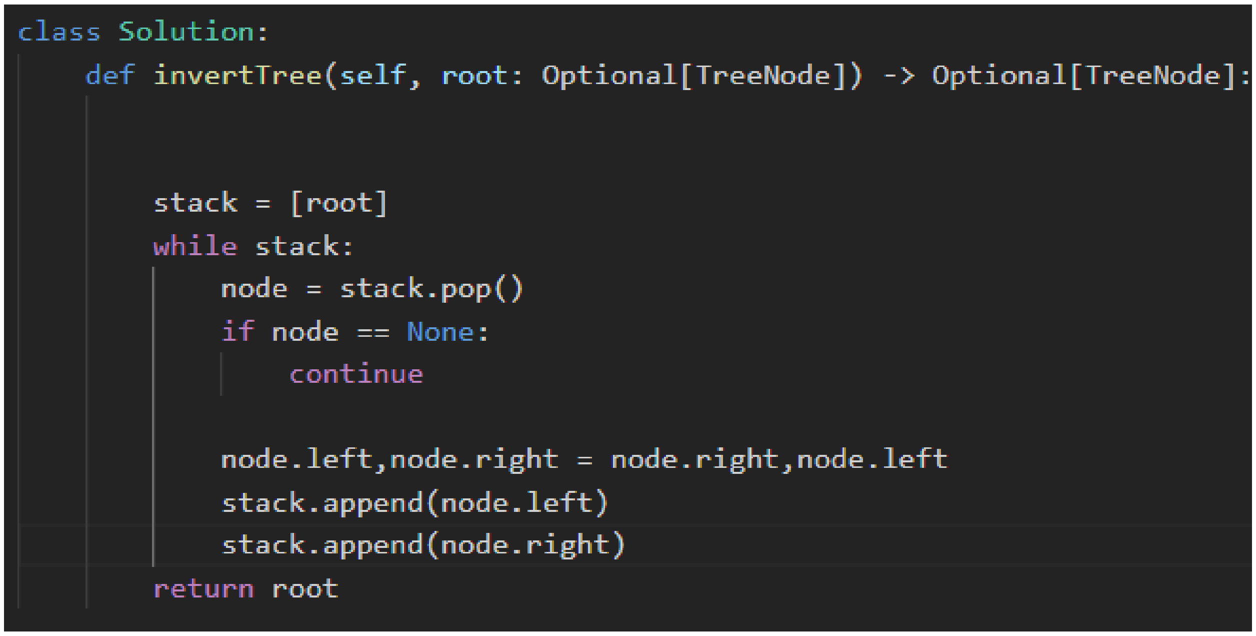
1. Given the `root` of a binary tree, invert the tree, and return *its root*.

   Example 1:

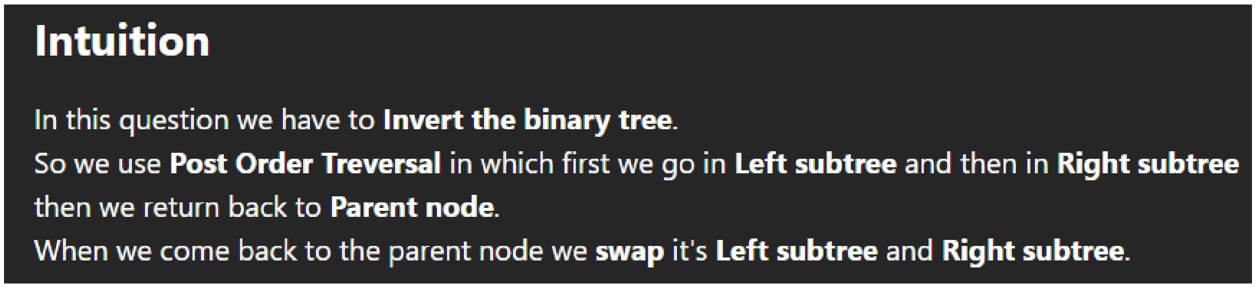
   ```
   Input: root = [4,2,7,1,3,6,9]
   Output: [4,7,2,9,6,3,1]
   ```

Approach using Stacks(self explanatory)

1. 

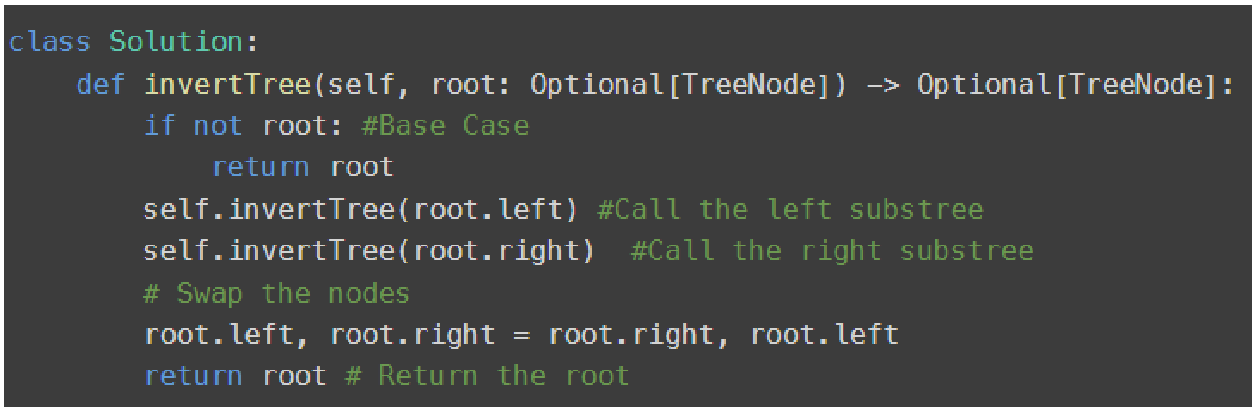
```python
class Solution:
    def invertTree(self, root: Optional[TreeNode]) -> Optional[TreeNode]:


        stack = [root]
        while stack:
            node = stack.pop()
            if node == None:
                continue

            node.left,node.right = node.right,node.left
            stack.append(node.left)
            stack.append(node.right)
        return root
```

Approach using Recursion

1. 

Intuition

In this question we have to **Invert the binary tree**.
So we use **Post Order Treversal** in which first we go in **Left subtree** and then in **Right subtree** then we return back to **Parent node**.
When we come back to the parent node we **swap** it's **Left subtree** and **Right subtree**.



```python
class Solution:
    def invertTree(self, root: Optional[TreeNode]) -> Optional[TreeNode]:
        if not root: #Base Case
            return root
        self.invertTree(root.left) #Call the left substree
        self.invertTree(root.right)  #Call the right substree
        # Swap the nodes
        root.left, root.right = root.right, root.left
        return root # Return the root
```