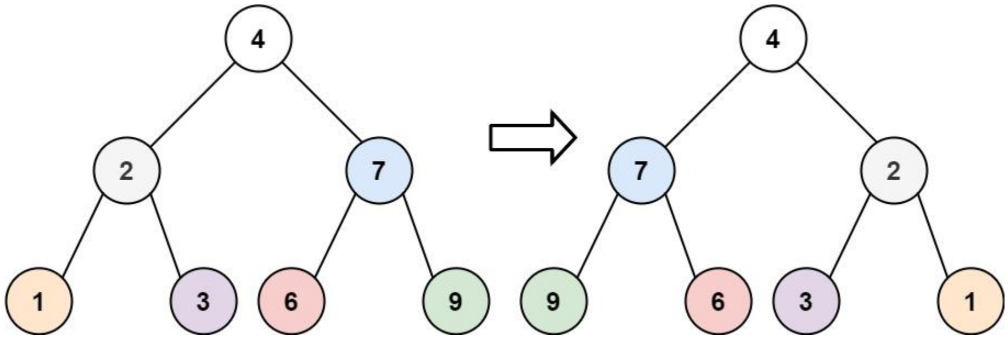
Invert Binary Tree

Question:

https://leetcode.com/problems/invert-binary-tree/submissions/

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 1:

My first approach was to use recursion to reverse the tree.

Solution 1:

https://gist.github.com/vermaayush680/4250336e428a4d881fb7ec1e79ef8edf

```
def invertTree1(self, root):
    if root:
        root.left, root.right = self.invertTree(root.right), self.invertTree(root.left)
        return root
```

Approach 2:

Using BFS + Queue to replace recursion.

Solution 2:

https://gist.github.com/vermaayush680/4b316352d2bb5957cbeea92f3af3c1c4

```
def invertTree2(self, root):
    queue = [(root)]
    while queue:
        node = queue.pop()
        if node:
            node.left, node.right = node.right, node.left
            queue.append(node.left)
            queue.append(node.right)
    return root
```

Approach 3:

Using DFS + Stack in place of BFS+Queue

Solution 3:

https://gist.github.com/vermaayush680/936b45a5304257428def01a2555d430a

```
def invertTree(self, root):
    stack = [root]
    while stack:
        node = stack.pop()
        if node:
            node.left, node.right = node.right, node.left
            stack.append(node.right)
            stack.append(node.left)
    return root
```