**Solve:**

| Python | Java |
| --- | --- |
| def Odd **/** EvenSwap(root):  def helper(node, level):  if not node:  return  ***#Set A***  if level % 2 == 1:  if node.left and node.right and   node.left.val <node.right.val:  node.left, node.right =   node.right, node.left  ***#Set B***  if level % 2 == 0:  if node.left and node.right and   node.right.val > node.left.val:  node.left, node.right =   node.right, node.left    helper(node.left, level + 1)  helper(node.right, level + 1)   helper(root, 0)  return root | public static BTNode Odd/EvenSwap(BTNode root) {  helper(root, 0);  return root;  }  private static void helper(BTNode node, int level) {  if (node == null) return;  ***//Set A***  if (level % 2 == 1) {  if (node.left != null && node.right != null &&   node.left.val < node.right.val) {  BTNode temp = node.left;  node.left = node.right;  node.right = temp;  }  }  ***// Set B***  if (level % 2 == 0) {  if (node.left != null && node.right != null &&   node.left.val < node.right.val) {  BTNode temp = node.left;  node.left = node.right;  node.right = temp;  }  }  helper(node.left, level + 1);  helper(node.right, level + 1);  } |

**Rubric:**

| **Part** | **Marks** |
| --- | --- |
| Correct Parameters main method and helper | 2 |
| Correct Base Case Handling | 2 |
| Checking the Correct Level and Value Comparison | 2 |
| Checking the Existence of Children Nodes | 3 |
| Correctly swapping | 2 |
| Correct Recursive call | 2 |
| Calling helper and return root | 2 |
|  | 15 |