SDE-SKILLS

January 11, 2020

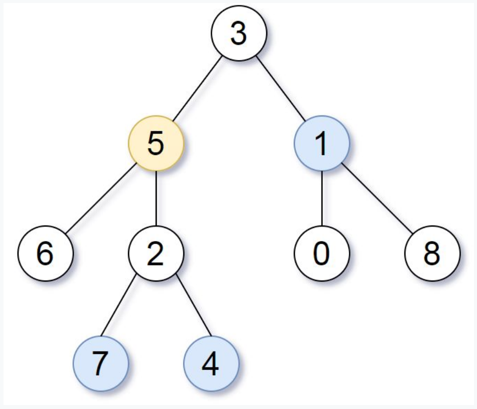
Topic: Binary Trees

[LC863]

We are given a binary tree (with root node `root`), a `target` node, and an integer value `K`. Return a list of the values of all nodes that have a distance `K` from the `target` node. The answer can be returned in any order.

```
Input: root = [3,5,1,6,2,0,8,null,null,7,4], target = 5, K = 2
```

```
Output: [7,4,1]
```



```
class Solution {
    public List<Integer> distanceK(TreeNode root, TreeNode target, int K) {

    }
}
```

[LC99] Recover Binary Search Tree

Two elements of a binary search tree (BST) are swapped by mistake. Recover the tree without changing its structure.

```
   1
  /
 3
  \
   2
```

```
Output: [3,1,null,null,2]
   3
  /
 1
  \
   2
class Solution {
    public void recoverTree(TreeNode root) {
    }
}
```

[LC 654] Given an integer array with no duplicates. A maximum tree building on this array is defined as follows:

1. **The root is the maximum number in the array.**
2. **The left subtree is the maximum tree constructed from left part subarray divided by the maximum number.**
3. **The right subtree is the maximum tree constructed from right part subarray divided by the maximum number.**

Construct the maximum tree by the given array and output the root node of this tree.

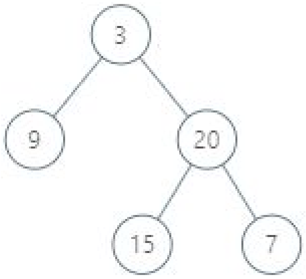
```
Input: [3,2,1,6,0,5]
Output: return the tree root node representing the following tree:
      6
    /   \
   3     5
    \    /
     2  0
       \
        1
```

```
class Solution {
    public TreeNode constructMaximumBinaryTree(int[] nums) {
}
}
```

[LC987] Vertical Order Traversal of a Binary Tree

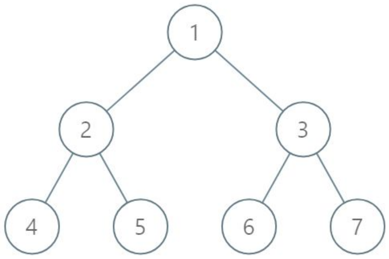
Given a binary tree, return the *vertical order* traversal of its nodes values.

Examples:



```
Input: [3,9,20,null,null,15,7]

Output: [[9],[3,15],[20],[7]]
```



```
Input: [1,2,3,4,5,6,7]

Output: [[4],[2],[1,5,6],[3],[7]]
```

```
class Solution {

    public List<List<Integer>> verticalTraversal(TreeNode root) {


    }

}
```