Binary Tree Maximum Path Sum

Given a binary tree, find the maximum path sum.

For this problem, a path is defined as any sequence of nodes from some starting node to any node in the tree along the parent-child connections. The path does not need to go through the root.

For example:
Given the below binary tree,

```
  1
 / \
2   3
```

Return `6`.

Solution 1

Here's my ideas:

- A path from start to end, goes up on the tree for 0 or more steps, then goes down for 0 or more steps. Once it goes down, it can't go up. Each path has a highest node, which is also the lowest common ancestor of all other nodes on the path.
- A recursive method `maxPathDown(TreeNode node)` (1) computes the maximum path sum with highest node is the input node, update maximum if necessary (2) returns the maximum sum of the path that can be extended to input node's parent.

Code:

```
public class Solution {
    int maxValue;

    public int maxPathSum(TreeNode root) {
        maxValue = Integer.MIN_VALUE;
        maxPathDown(root);
        return maxValue;
    }

    private int maxPathDown(TreeNode node) {
        if (node == null) return 0;
        int left = Math.max(0, maxPathDown(node.left));
        int right = Math.max(0, maxPathDown(node.right));
        maxValue = Math.max(maxValue, left + right + node.val);
        return Math.max(left, right) + node.val;
    }
}
```

written by wei-bung original link here

Solution 2

```
class Solution {
    int maxToRoot(TreeNode *root, int &re) {
        if (!root) return 0;
        int l = maxToRoot(root->left, re);
        int r = maxToRoot(root->right, re);
        if (l < 0) l = 0;
        if (r < 0) r = 0;
        if (l + r + root->val > re) re = l + r + root->val;
        return root->val += max(l, r);
    }
public:
    int maxPathSum(TreeNode *root) {
        int max = -2147483648;
        maxToRoot(root, max);
        return max;
    }
};
```

update the val of each node of the tree bottom-up, the new val of TreeNode *x stands for the max sum started from any node in subtree x and ended in x, mataining the re for result in traversal at the same time.

written by xt2357 original link here

Solution 3

```
public class Solution {
    int max = Integer.MIN_VALUE;

    public int maxPathSum(TreeNode root) {
        helper(root);
        return max;
    }

    // helper returns the max branch
    // plus current node's value
    int helper(TreeNode root) {
        if (root == null) return 0;

        int left = Math.max(helper(root.left), 0);
        int right = Math.max(helper(root.right), 0);

        max = Math.max(max, root.val + left + right);

        return root.val + Math.max(left, right);
    }
}
```

written by jeantimex original link here