Maximum Sum of Nodes

Problem statement Send feedback

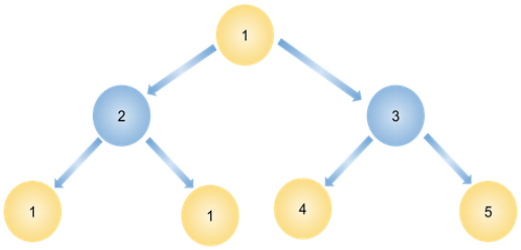
You have been given a binary tree with an integer value associated to each node. You are supposed to choose a subset of these nodes such that the sum of these chosen nodes is maximum. Keep in mind that no two of the chosen nodes must be adjacent to each other.

Note :

Two nodes are said to be adjacent to each other if they are directly connected to each other. This means that if a node is taken as part of the sum, then none of its children can be considered for the same and vice versa.

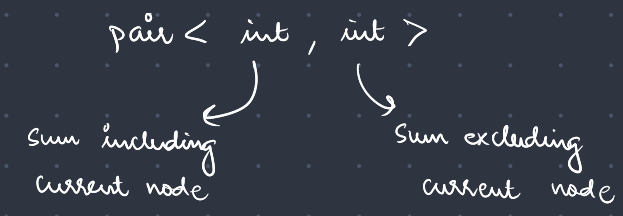
For example :

For the given binary tree



Nodes used in consideration for maximum sum such that no two of them are adjacent are highlighted. Maximum sum of nodes = 1 + 1 + 1 + 4 + 5 = 12.

→ Approach:




```
pair<int, int> solve(TreeNode<int>* root) {
    if(root == NULL) {
        return make_pair(0,0);
    }

    pair<int, int> left = solve(root->left);
    pair<int, int> right = solve(root->right);

    pair<int, int> res;
    res.first = root->data + left.second + right.second;
    res.second = max(left.second, left.first) + max(right.second, right.first);
    return res;
}

int maximumSumOfNodes(TreeNode<int> *root)
{
    pair<int, int> answer = solve(root);
    return max(answer.first, answer.second);
}
```

Time Complexity : O(N)

Space Complexity : O(N)

Can also be solved using Memoization