**SOLUTION 3**

#include <iostream>

#include <climits>

using namespace std;

class TreeNode {

public:

int val;

TreeNode\* left;

TreeNode\* right;

TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

};

class Solution {

public:

int maxPathSum(TreeNode\* root) {

int maxSum = INT\_MIN;

maxGain(root, maxSum);

return maxSum;

}

private:

int maxGain(TreeNode\* node, int& maxSum) {

if (node == nullptr) {

return 0;

}

int leftGain = max(maxGain(node->left, maxSum), 0);

int rightGain = max(maxGain(node->right, maxSum), 0);

int currentSum = node->val + leftGain + rightGain;

maxSum = max(maxSum, currentSum);

return node->val + max(leftGain, rightGain);

}

};

int main() {

TreeNode\* root = new TreeNode(1);

root->left = new TreeNode(2);

root->right = new TreeNode(3);

root->left->left = new TreeNode(4);

root->left->right = new TreeNode(5);

Solution sol;

cout << "The maximum path sum is: " << sol.maxPathSum(root) << endl;

// Clean up memory

delete root->left->right;

delete root->left->left;

delete root->right;

delete root->left;

delete root;

return 0;

} 