Задача №1

Путь в бинарном дереве — это последовательность узлов, в которой каждая пара соседних узлов в последовательности имеет соединяющее их ребро . Узел может появиться в последовательности не более одного раза . Обратите внимание, что путь не обязательно должен проходить через корень. Сумма путей пути — это сумма значений узлов в пути. Учитывая rootдвоичное дерево, верните максимальную сумму путей любого непустого пути.
Пример 1: Входные данные: root = [1,2,3] Выходные данные: 6 Объяснение: Оптимальный путь — 2 -> 1 -> 3 с суммой путей 2 + 1 + 3 = 6.
Пример 2: Ввод: root = [-10,9,20,null,null,15,7] Выход: 42 Объяснение: Оптимальный путь — 15 -> 20 -> 7 с суммой путей 15 + 20 + 7 = 42.
Ограничения: Количество узлов в дереве находится в диапазоне .[1, 3 * 104] -1000 <= Node.val <= 1000

```
#include <iostream>
#include <algorithm>

using namespace std;

struct TreeNode {
    int val;
    TreeNode* left;
    TreeNode* right;
    TreeNode(int x) : val(x), left(NULL), right(NULL) {}
};

int maxPathSum(TreeNode* root, int& maxSum) {
    if (root == NULL) {
        return 0;
    }

    int leftSum = maxPathSum(root->left, maxSum);
    int rightSum = maxPathSum(root->right, maxSum);
    int currentSum = root->val + max(0, leftSum) + max(0, rightSum);
    maxSum = max(maxSum, currentSum);
    return root->val + max(0, max(leftSum, rightSum));
}

TreeNode* createTestTree() {
    TreeNode* root = new TreeNode(7);
    root->left = new TreeNode(5);
    root->right = new TreeNode(8);
    return root;
}

int main() {
    TreeNode* root = createTestTree();
    int maxSum = INT_MIN;
    maxPathSum(root, maxSum);
    cout << "Max lenght: " << maxSum << endl;

    return 0;
}
```

```
Max lenght: 20


** Process exited - Return Code: 0 **
```