

```

package binarytree;

// Java program to find maximum path sum in Binary Tree

// An object of Res is passed around so that the
// same value can be used by multiple recursive calls.
class Res {
    public int val;
}

class BinaryTree69 {

    // Root of the Binary Tree
    Node root;

    // This function returns overall maximum path sum in 'res'
    // And returns max path sum going through root.
    int findMaxUtil(Node node, Res res)
    {
        // Base Case
        if (node == null)
            return 0;

        // l and r store maximum path sum going through left and
        // right child of root respectively
        int l = findMaxUtil(node.left, res);
        int r = findMaxUtil(node.right, res);

        // Max path for parent call of root. This path must
        // include at-most one child of root
        int max_single = Math.max(Math.max(l, r) + node.data,
                                   node.data);

        // Max Top represents the sum when the Node under
        // consideration is the root of the maxsum path and no
        // ancestors of root are there in max sum path
        int max_top = Math.max(max_single, l + r + node.data);

        // Store the Maximum Result.
        res.val = Math.max(res.val, max_top);

        return max_single;
    }

    int findMaxSum() {
        return findMaxSum(root);
    }

    // Returns maximum path sum in tree with given root
    int findMaxSum(Node node) {

        // Initialize result
        // int res2 = Integer.MIN_VALUE;
        Res res = new Res();
        res.val = Integer.MIN_VALUE;

        // Compute and return result
        findMaxUtil(node, res);
        return res.val;
    }

    /* Driver program to test above functions */
    public static void main(String args[]) {

```

```
BinaryTree69 tree = new BinaryTree69();  
tree.root = new Node(10);  
tree.root.left = new Node(2);  
tree.root.right = new Node(10);  
tree.root.left.left = new Node(20);  
tree.root.left.right = new Node(1);  
tree.root.right.right = new Node(-25);  
tree.root.right.right.left = new Node(3);  
tree.root.right.right.right = new Node(4);  
System.out.println("maximum path sum is : " +  
    tree.findMaxSum());
```