

# Problem 1315: Sum of Nodes with Even-Valued Grandparent

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 85.83%

**Paid Only:** No

**Tags:** Tree, Depth-First Search, Breadth-First Search, Binary Tree

## Problem Description

Given the `root` of a binary tree, return \_the sum of values of nodes with an\*\*even-valued grandparent\*\*\_. If there are no nodes with an \*\*even-valued grandparent\*\* , return `0`.

A \*\*grandparent\*\* of a node is the parent of its parent if it exists.

**Example 1:**



**Input:** root = [6,7,8,2,7,1,3,9,null,1,4,null,null,null,5] **Output:** 18 **Explanation:** The red nodes are the nodes with even-value grandparent while the blue nodes are the even-value grandparents.

**Example 2:**



**Input:** root = [1] **Output:** 0

**Constraints:**

\* The number of nodes in the tree is in the range `[1, 104]`. \* `1 <= Node.val <= 100`

## Code Snippets

### C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
 *     right(right) {}
 * };
 */
class Solution {
public:
    int sumEvenGrandparent(TreeNode* root) {

    }
};

}
```

### Java:

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {
    public int sumEvenGrandparent(TreeNode root) {
```

```
    }  
    }
```

### Python3:

```
# Definition for a binary tree node.  
# class TreeNode:  
#     def __init__(self, val=0, left=None, right=None):  
#         self.val = val  
#         self.left = left  
#         self.right = right  
class Solution:  
    def sumEvenGrandparent(self, root: Optional[TreeNode]) -> int:  
        pass
```