

```
1  /*
2  给你一棵二叉树，请你返回满足以下条件的所有节点的值之和：
3
4      该节点的祖父节点的值偶数。（一个节点的祖父节点是指该节点的父节点的父节点。）
5
6  如果不存在祖父节点值为偶数的节点，那么返回 0 。
7
8
9
10 示例：
11
12 输入：root = [6,7,8,2,7,1,3,9,null,1,4,null,null,null,5]
13 输出：18
14 解释：图中红色节点的祖父节点的值偶数，蓝色节点为这些红色节点的祖父节点。
15
16
17
18 提示：
19
20     树中节点的数目在 1 到 10^4 之间。
21     每个节点的值在 1 到 100 之间。
22
23 来源：力扣（LeetCode）
24 链接：https://leetcode-cn.com/problems/sum-of-nodes-with-even-valued-grandparent
25 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
26 */
```

分析：

- DFS遇到值为偶数的节点后，遍历该节点的孙子节点即可。

方法一：C++_DFS

```
1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9   */
10 class Solution
11 {
12     private:
13         int ret_val = 0;
14
15 }
```

```

16 void travelGrandson(TreeNode* node)
17 {
18     if(node==NULL)
19     {
20         return;
21     }
22
23     if(node->left)
24     {
25         ret_val += node->left->val;
26     }
27
28     if(node->right)
29     {
30         ret_val += node->right->val;
31     }
32
33
34 }
35
36 void dfs(TreeNode* node )
37 {
38     if(node==NULL)
39     {
40         return;
41     }
42
43     if(node->val%2==0)
44     {
45         travelGrandson(node->left);
46         travelGrandson(node->right);
47     }
48
49
50     dfs(node->left);
51     dfs(node->right);
52 }
53
54
55
56 public:
57 int sumEvenGrandparent(TreeNode* root)
58 {
59     ret_val = 0;
60     dfs(root);
61     return ret_val;
62
63 }
64 };

```

66 /*

67 执行结果:

68 通过

69 显示详情

70 执行用时 :44 ms, 在所有 C++ 提交中击败了94.30% 的用户

71 内存消耗 :31.7 MB, 在所有 C++ 提交中击败了100.00%的用户

72 */

