# 题目

给定一个二叉树，找到最长的路径，这个路径中的每个节点具有相同值。 这条路径可以经过也可以不经过根节点。

注意：两个节点之间的路径长度由它们之间的边数表示。

**示例 1:**

输入:

5

/ \

4 5

/ \ \

1 1 5

输出:

2

**示例 2:**

输入:

1

/ \

4 5

/ \ \

4 4 5

输出:

2

注意: 给定的二叉树不超过10000个结点。 树的高度不超过1000。

# 分析

## 方法一：递归法

代码：

class Solution {

public:

    int res;

    int longestUnivaluePath(TreeNode \*root)

    {

        if (!root)

            return 0;

        longestPath(root);

        return res;

    }

    int longestPath(TreeNode \*root)

    {

        if (!root)

            return 0;

        int left = longestPath(root->left), right = longestPath(root->right);

        // 如果存在左子节点和根节点同值，更新左最长路径;否则左最长路径为0

        if (root->left && root->val == root->left->val)

            left++;

        else

            left = 0;

        if (root->right && root->val == root->right->val)

            right++;

        else

            right = 0;

        res = max(res, left + right);

        return max(left, right);

    }

};

**另一种写法：**

class Solution {

public:

int dfs(TreeNode\* root,int& res)

{

if(!root)

return 0;

int l=0,r=0;

if(root->left)

{

int tmp = dfs(root->left,res);

if(root->val == root->left->val)

l = tmp+1;

}

if(root->right)

{

int tmp = dfs(root->right,res);

if(root->val == root->right->val)

r = tmp+1;

}

res = res>(r+l)?res:r+l;

return l>r?l:r;

}

int longestUnivaluePath(TreeNode\* root) {

int res = 0;

dfs(root,res);

return res;

}

};