

在一排树中，第 i 棵树产生 $tree[i]$ 型的水果。

你可以从你选择的任何树开始，然后重复执行以下步骤：

1. 把这棵树上的水果放进你的篮子里。如果你做不到，就停下来。
2. 移动到当前树右侧的下一棵树。如果右边没有树，就停下来。

请注意，在选择一颗树后，你没有任何选择：你必须执行步骤 1，然后执行步骤 2，然后返回步骤 1，然后执行步骤 2，依此类推，直至停止。

你有两个篮子，每个篮子可以携带任何数量的水果，但你希望每个篮子只携带一种类型的水果。

用这个程序你能收集的水果总量是多少？

示例 1:

输入: $[1, 2, 1]$

输出: 3

解释: 我们可以收集 $[1, 2, 1]$ 。

示例 2:

输入: $[0, 1, 2, 2]$

输出: 3

解释: 我们可以收集 $[1, 2, 2]$ 。

如果我们从第一棵树开始，我们将只能收集到 $[0, 1]$ 。

示例 3:

输入: $[1, 2, 3, 2, 2]$

输出: 4

解释: 我们可以收集 $[2, 3, 2, 2]$ 。

如果我们从第一棵树开始，我们将只能收集到 $[1, 2]$ 。

示例 4:

输入: $[3, 3, 3, 1, 2, 1, 1, 2, 3, 3, 4]$

输出: 5

解释: 我们可以收集 $[1, 2, 1, 1, 2]$ 。

如果我们从第一棵树或第八棵树开始，我们将只能收集到 4 个水果。

提示:

1. $1 \leq tree.length \leq 40000$
2. $0 \leq tree[i] < tree.length$

思路，该题为双指针的题，一个指针记录其不同的位置j，一个指针进行数组的遍历，当数组包含两个元素时，遇到第三个不同的元素，数组长度进行变化，第三个数组长度为前一个数组最靠近他的连续元素和与自己相加。像1 2 2 3 3 2 2中，遍历到 1 2 2时，2个元素数组为1 2 2，遍历到3时，数组变为 2 2 3其中3是与1进行比较，确定元素是不是为两个。

代码

```
class Solution {
public:
    int totalFruit(vector<int>& tree) {
        if(tree.size()==0){
            return 0;
        }
        bool flag=false;
        int j=0;
        int max=1;
        int num=1;
        for(int i=1;i<tree.size();i++){
            if(tree[i]!=tree[i-1]){
                if(tree[i]==tree[j]&&flag){
                    num++;
                    j=i-1;
                }else{
                    if(flag==false){
                        j=-1;
                    }
                    flag=true;
                    num=i-j;
                    j=i-1;
                }
            }else{
                num++;
            }
            if(num>max){
                max=num;
            }
        }
    }
};
```

```
    }  
    return max;  
}  
};
```