

分糖果

给定一个偶数长度的数组，其中不同的数字代表着不同种类的糖果，每一个数字代表一个糖果。你需要把这些糖果平均分给一个弟弟和一个妹妹。返回妹妹可以获得的最大糖果的种类数。

示例 1:

输入: candies = [1,1,2,2,3,3]

输出: 3

解析: 一共有三种种类的糖果，每一种都有两个。

最优分配方案: 妹妹获得[1,2,3], 弟弟也获得[1,2,3]。这样使妹妹获得糖果的种类数最多。

示例 2:

输入: candies = [1,1,2,3]

输出: 2

解析: 妹妹获得糖果[2,3], 弟弟获得糖果[1,1], 妹妹有两种不同的糖果，弟弟只有一种。这样使得妹妹可以获得的糖果种类数最多。

```
class Solution {
public:
    int distributeCandies(vector<int>& candyType) {
        /*
        unordered_set<int> map;
        for(int i=0;i<candyType.size();i++)
        {
            map.insert(candyType[i]);
        }
        */
        int cnt=1;
        sort(candyType.begin(),candyType.end());
        for(int i=0;i<candyType.size()-1;i++)
        {
            if(candyType[i]!=candyType[i+1])
            {
                cnt++;
            }
        }
        /*
        只要统计糖果种类数 cnt 即可。
        当 cnt >= A.length/2 时，每种拿出 1 个给妹妹，就足够了。
        */
    }
};
```

```
    当  $\text{cnt} < A.\text{length}/2$  时, 最多  $\text{cnt}$  种  
    */  
    if( $\text{cnt} \geq \text{candyType}.\text{size}()/2$ )  
        return  $\text{candyType}.\text{size}()/2$ ;  
    else  
        return cnt;  
}  
};
```