# 题目

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。

注意:

数组的长度为[2, 10,000]，并且确定为偶数。

数组中数字的大小在范围[-100,000, 100,000]内。

# 分析

## 方法一：暴力法

## 方法二：哈希

**思路：**

通过哈希法，可以求出了糖果的类型数量，如果糖果种类大于糖果总数的一半了，返回 糖果数量的一半就好，因为妹妹已经得到种类最多的糖果了，否则，就是返回糖果的种类。

**代码：**

class Solution {

public:

int distributeCandies(vector<int>& candies) {

// 初始化一个record数组，因为糖果种类的数值在范围[-100,000, 100,000]内

// 将这个范围的数值统一加上100000，可以映射到record数组的索引下表了

// record数组大小必须大于等于200001，这样才能取到200000这个下表索引

int record[200001] = {0};

// 通过record来记录糖果的种类

for (int i = 0; i < candies.size(); i++) {

record[candies[i] + 100000]++;

}

// 统计糖果种类的数量

int count = 0;

for (int i = 0; i < 200001; i++) {

if (record[i] != 0) {

count ++;

}

}

int half = candies.size() / 2;

// 如果糖果种类大于糖果总数的一半了，return 糖果数量的一半

// 否则，就是return 糖果的种类。

return count > half ? half : count;

}

};

## 方法三：unordered\_set

**思路：**

既然平均分，那么就是分总数的一半，那么如果我的种类数多于一半了，肯定就要返回总数的一半，如果不多余，则返回种类数给妹妹。

**代码：**

class Solution {

public:

int distributeCandies(vector<int>& candies) {

unordered\_set<int> Set;

for (auto i:candies)

Set.insert(i);

return min(Set.size(), candies.size() / 2);

}

};

另一种写法：

class Solution {

public:

    int distributeCandies(vector<int>& candyType) {

        unordered\_set<int> set;

        for(auto item : candyType)

        {

            set.insert(item);

        }

        if(set.size()<=candyType.size()/2)

            return set.size();

        return candyType.size()/2;

    }

};