575 Distribute Candies (link)

Description

Alice has `n` candies, where the i^{th} candy is of type `candyType[i]`. Alice noticed that she started to gain weight, so she visited a doctor.

The doctor advised Alice to only eat `n / 2` of the candies she has (`n` is always even). Alice likes her candies very much, and she wants to eat the maximum number of different types of candies while still following the doctor's advice.

Given the integer array `candyType` of length `n`, return *the **maximum** number of different types of candies she can eat if she only eats* `n / 2` *of them*.

Example 1:

```
Input: candyType = [1,1,2,2,3,3]
Output: 3
Explanation: Alice can only eat 6 / 2 = 3 candies. Since there are only 3 types, she
```

Example 2:

```
Input: candyType = [1,1,2,3]
Output: 2
Explanation: Alice can only eat 4 / 2 = 2 candies. Whether she eats types [1,2], [1,3
```

Example 3:

```
Input: candyType = [6,6,6,6]
Output: 1
Explanation: Alice can only eat 4 / 2 = 2 candies. Even though she can eat 2 candies,
```

Constraints:

- `n == candyType.length`
- $2 <= n <= 10^4$
- `n` is even.
- $-10^5 <= candyType[i] <= 10^5$

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
#include <unordered_set>

using namespace std;

class Solution {
public:
    int distributeCandies(vector<int>& candyType) {
        unordered_set<int> types;


        for (int candy : candyType) {
            types.insert(candy);
        }


        int maxTypes = min(types.size(), candyType.size() / 2);

        return maxTypes;
    }
};
```