Assignment 2:

1. 561. Array Partition

   Given an integer array nums of 2n integers, group these integers into n pairs (a1, b1), (a2, b2), ..., (an, bn) such that the sum of min(ai, bi) for all i is maximized. Return the maximized sum.

```
//Time Complexity: O(n)

//Space Complexity:O(1)

class Solution {

public:

    int arrayPairSum(vector<int>& nums) {

        //[1,4,3,2]

        //1. Sort [1,2,3,4]

        //2. Optimized Pair (1,2)+(3,4)

        //3. Sum min values: 1+3=4

        sort(nums.begin(),nums.end());

        int sum=0;

        for(int i=0;i<nums.size();i+=2)

        {

            sum=sum+min(nums[i],nums[i

            +1]);

        }

        return sum;

    }

};
```

2. 575. Distribute Candies

Alice has n candies, where the ith 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.

```
//Time Complexity: O(n)

//Space Complexity: O(1)


class Solution {

public:

    int distributeCandies(vector<int>& candyType) {

        set<int> s(candyType.begin(),candyType.end());

        return min(s.size(),candyType.size()/2);


    }

};
```