

解法一 改变原数组，排序

解法二 哈希表

解法三 交换

题目一：找出数组中重复的数字。

在一个长度为 n 的数组里的所有数字都在 $0 \sim n-1$ 的范围内。数组中某些数字是重复的，但不知道有几个数字重复了，也不知道每个数字重复了几次。请找出数组中任意一个重复的数字。例如，如果输入长度为 7 的数组 {2, 3, 1, 0, 2, 5, 3}，那么对应的输出是重复的数字 2 或者 3。

解法一 改变原数组，排序

时间复杂度 $O(n \log n)$ ，空间 $O(1)$

```
int getDuplication(int *numbers,int length)
{
    sort(numbers,numbers+length);
    for(int i=1;i<length;i++)
    {
        if(numbers[i]==numbers[i-1])
            return numbers[i];
    }
}
```

解法二 哈希表

时间 $O(n)$ ，空间 $O(n)$

```
int getDuplication(vector<int> nums)
{
    if(nums.empty())
        return -1;
    int n=nums.size();
    unordered_map<int,int> m(n,0);
    for(int i=0;i<nums.size();i++)
    {
        if(m[nums[i]]==1)
            return nums[i];
    }
}
```

```
        m[nums[i]]++;
    }
    return -1;
}
```

解法三 交换

时间 $O(n)$ ，空间 $O(1)$

如果当前的数字不等于当前的位置下标，就把当前数字交换到其对应的位置上去，循环比较和交换，直到发现重复数字

```
int getDuplication(vector<int> nums)
{
    for(int i=0;i<nums.size();i++)
    {
        while(nums[i]!=i)
        {
            if(nums[i]==nums[nums[i]])
                return nums[i];
            swap(nums[i],nums[nums[i]]);
        }
    }
}
```