# 217 Contains Duplicate

2018年4月4日 21:06

## Question:

Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

来自 <https://leetcode.com/problems/contains-duplicate/description/>

给定一个整数数组，判断是否存在重复元素。
如果任何值在数组中出现至少两次，函数应该返回 true。如果每个元素都不相同，则返回 false。

## Solution for Python3:

```python
class Solution1:
    def containsDuplicate(self, nums):
        """
        :type nums: List[int]
        :rtype: bool
        """
        nums.sort()
        for i in range(1, len(nums)):
           if nums[i] == nums[i - 1]:
               return True;
        return False

class Solution2:
    def containsDuplicate(self, nums):
        """
        :type nums: List[int]
        :rtype: bool
        """
        s = set()
        for num in nums:
           if  num in s:
               return True
           s.add(num)
        return False

class Solution3:
    def containsDuplicate(self, nums):
        """
        :type nums: List[int]
        :rtype: bool
        """
        return len(nums) > len(set(nums))
```

## Solution for C++:

```cpp
// set
class Solution1 {
public:
    bool containsDuplicate(vector<int>& nums) {
        set<int> s;
        for (int num : nums) {
            if (!s.insert(num).second) {
                return true;
            }
        }
        return false;
    }
};

class Solution2 {
public:
    bool containsDuplicate(vector<int>& nums) {
        set<int> s;
        for (int num : nums) {
            if (s.count(num)) {
                return true;
            }
            s.insert(num);
        }
        return false;
    }
};

class Solution3 {
public:
    bool containsDuplicate(vector<int>& nums) {
        set<int> s;
        for (int num : nums) {
            if (s.find(num) != s.end()) {
                return true;
            } else {
                s.insert(num);
            }
        }
        return false;
    }
}

class Solution4 {
public:
    bool containsDuplicate(vector<int>& nums) {
        return nums.size() > unordered_set<int>(nums.begin(), nums.end()).size();
    }
}

// sort 方法
class Solution5 {
public:
    bool containsDuplicate(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        for (int i = 0; i < nums.size() - 1 ; i++) {
            if (nums[i] == nums[i + 1]) {
                return true;
            }
        }
    }
}
```

**Appendix:**

**Python sort(iterable, index=0):**

1) Sorted(iterable, cmp, key, reverse)
2) Iterable.sort(cmp, key, reverse)
3) iterable指定要排序的list或者iterable
4) cmp为比较大小的函数
5) key为函数，指定待取的元素项
6) reverse默认False(升序)

**Python set:**

1) 创建：set1 = {'a','b'}  set2 = set(list)
2) 添加：s = set()  s.add(1)  s.update(list)将list拆分元素传入
3) 清空：s.clear()
4) 删除：s.remove(1)  若元素不存在会报错
5) 移除：s.discard(1)  若元素不存在不会报错
6) 随机移除：ss = s.pop() 随机移除某个元素并返回
7) 判断有误交集：s1.isdidjoint(s2),无交集返回True
8) 交集： s1.intersection(s2)
9) 并集：s1.union(s2)
10) 对称差集：s1.symmetric_difference(s2)
11) 差集：s1.difference(s2)  s1中存在，s2中不存在S