219 Contains Duplicate II

2018年4月4日 21:06

Question:

Given an array of integers and an integer *k*, find out whether there are two distinct indices *i* and *j* in the array such that **nums[i] = nums[j]** and the **absolute** difference between *i* and *j* is at most *k*.

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

给定一个整数数组和一个整数 *k*，判断数组中是否存在两个不同的索引 *i* 和 *j*，使 nums [i] = nums [j]，并且 *i* 和 *j* 的绝对差值最大为 *k*。

Solution for Python3:

```python
class Solution1:
    def containsNearbyDuplicate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: bool
        """
        s = set()
        for i in range(len(nums)):
            if i > k:
                s.discard(nums[i - k - 1])
            if nums[i] in s:
                return True
            s.add(nums[i])
        return False

class Solution2:
    def containsNearbyDuplicate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: bool
        """
        dic = {}
        for i, v in enumerate(nums):
            if v in dic and i - dic[v] <= k:
                return True
            dic[v] = i
        return False
```

Solution for C++:

```cpp
class Solution {
public:
    bool containsNearbyDuplicate(vector<int>& nums, int k) {
        unordered_set<int> s;
        for (int i = 0; i < nums.size(); i++) {
            if (i > k) {
```

```
                s.erase(nums[i - k - 1]);
            }
            if (!s.insert(nums[i]).second) {
                return true;
            }
        }
        return false;
    }
};
```