

## 219. Contains Duplicate II

Easy



5.4K



2.8K



Companies

Given an integer array `nums` and an integer `k`, return `true` if there are two **distinct indices** `i` and `j` in the array such that `nums[i] == nums[j]` and `abs(i - j) <= k`.

### Example 1:

**Input:** `nums = [1,2,3,1]`, `k = 3`

**Output:** `true`

### Example 2:

**Input:** `nums = [1,0,1,1]`, `k = 1`

**Output:** `true`

### Example 3:

**Input:** `nums = [1,2,3,1,2,3]`, `k = 2`

**Output:** `false`

### Constraints:

- `1 <= nums.length <= 105`
- `-109 <= nums[i] <= 109`
- `0 <= k <= 105`

Accepted **736.4K**

Submissions **1.7M**

Acceptance Rate **43.2%**

Bruteforce:

$$T(n) = O(n^2)$$

$$S(n) = O(1)$$

Approach 1: Using hashman

Approach 1: Using hashmap

`unordered_map<int, int> m;`

value  $\swarrow$   $\searrow$  index

$T(n) : O(n)$

$S(n) : O(n)$

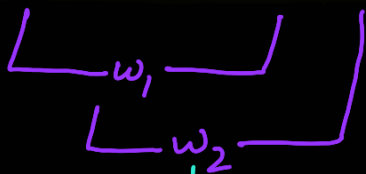
Approach 2:

Using sliding window technique along with hashmap we can reduce the space complexity from

$O(n) \rightarrow O(k+1)$

i.e. at any given point of time the hashmap contains only  $k$  elements.

Note: we can also use  
hashset in place of  
hashmap.



here in this window we found that duplicate element &  $4 - 1 = 3 \leq 3$

Q: But here the main catch is how are we going to implement this logic in code?

→ Take two pointers  $l$  &  $r$   
where  $l$  is pointer to start of window  
and  $r$  is pointer to end of window.

→ Run a loop until  $r$  reaches end of array i.e. last window encountered.

→ at any given point of time if  
 $windowSize > k$

Then remove the leftmost element of window & slide the window by incrementing  $l$ .

→ after the window is adjusted see if element at  $i^{th}$  index (i.e. element pointed by  $r$  pointer) is already there in window or not. If it is return true else add it to the window.

9 / 8 / 6 / 8 / 3 / 4

$l=0$       2. map  $m = \{ \}$

$r=0$

→  $r=0$

i)  $r$  is not greater than  $k$ .

ii)  $nums[r]$  is not in the window

iii) So add it.

$\{ < 9, 1 > \}$

→  $r=1$

i)  $r$  is not greater than  $k$ .

ii)  $nums[r]$  is not in window

iii) So add it

$\{ < 9, 1 > ,$

$< 3, 1 > \}$

→  $x = 2$

- i)  $x$  is not greater than  $k$ .
- ii)  $\text{nums}[x]$  is not in window
- iii) So add it

$\{ \langle 9, 1 \rangle$   
 $\langle 3, 1 \rangle$   
 $\langle 6, 1 \rangle \}$

→  $x = 3$

- i)  $x$  is not greater than  $k$ .
- ii)  $\text{nums}[x]$  is not in window
- iii) So add it

$\{ \langle 9, 1 \rangle$   
 $\langle 3, 1 \rangle$   
 $\langle 6, 1 \rangle$   
 $\langle 8, 1 \rangle \}$

→  $x = 4$

- i)  $x$  is greater than  $k$ .  
So remove  $\text{nums}[l]$  from map

$\{ \langle 3, 1 \rangle$   
 $\langle 6, 1 \rangle$   
 $\langle 8, 1 \rangle \}$

- ii)  $\text{nums}[r]$  is in window  
So return true.