Kth Distance

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Hash Map set
language	C++
solved on	@02/11/2024
link	https://www.geeksforgeeks.org/problems/kth-distance3757/1
Completion	☑

Intuition

To check for duplicates within a window of size `k` in an array, we need to use a data structure that allows quick insertions, deletions, and lookups. The best choice for this is an `unordered_set`, which provides `O(1)` average time complexity for these operations. The idea is to keep a sliding window of size `k` while iterating over the array and check for duplicates within this window.

Approach

1. Initialize an `unordered_set` to keep track of elements within the current window of size `k`.
2. Iterate through the first `k` elements of the array and add them to the set.
3. If the size of the set is less than `k`, it means a duplicate was found within the first `k` elements, so return `true`.
4. Iterate over the remaining elements in the array from index `k` to `arr.size() - 1`:
    - Check if the current element exists in the set:
        - If it does, return `true` (duplicate found).
    - Remove the element that is moving out of the window (`arr[j - k]`).
    - Insert the current element (`arr[j]`) into the set.
5. If the loop completes without finding duplicates, return `false`.

Complexity

Time Complexity:

- **O(n)**: We iterate through the array once, and each insertion, deletion, and lookup operation in an `unordered_set` takes `O(1)` on average.

Space Complexity:

- **O(k)**: The space required by the `unordered_set` is at most `k`, as it holds up to `k` elements at any time.

Code

```
class Solution {
  public:
    bool checkDuplicatesWithinK(vector<int>& arr, int k) {
        unordered_set<int> mySet;
        for(int i = 0; i<k; i++) mySet.insert(arr[i]);
        if(mySet.size()<k) return true;
        for(int j = k; j < arr.size(); j++){
            if(mySet.find(arr[j])!=mySet.end()) return true;
            mySet.erase(arr[j-k]);
            mySet.insert(arr[j]);
        }
        return false;
    }
};
```