**Check if the following given array contains duplicate elements within k distance from each other.**

Given an unsorted array that may contain duplicates. Also given a number k which is smaller than the size of the array, returns true if the array contains duplicates within k distance.

**Example-1:**

**Input:** k = 3, arr[] = {1, 2, 3, 4, 1, 2, 3, 4}

**Output:**false

All duplicates are more than k(3) distance away.

1...1(has a distance of 4)

2...2(has a distance of 4)

3...3 and 4...4 are similar

Even if we find one duplicate number having a distance less than equal to k(<=k), we would return true.

**Example-2:**

**Input:** k = 3, arr[] = {1, 2, 3, 1, 4, 5}

**Output:** true

1 is repeated at distance 3(3<=k hence we return true).

private static boolean kDistance(int [] nums , int k){

int n = nums.length();

//hash Map to store frequencies

Map<Integer , Integer > map = new HashMap<>();

for(int i = 0 ; i < nums.length ; i++){

/\*

\* Check if the number is present in the Map or Not

\* If present then check if it is less than or equal at k-disatnce

\*If so , return true

\*/

if(map.containsKey(nums[i]) && i - map.get(nums[i]) <= k)

return true;

//For each iteration , update the position of the element

map.put(nums[i] , i);

}

return false;

}