217. Contains Duplicate

Problem Statement

Check the problem statement here.

My Solution

1. Basic which is not efficient and Time Limit Exceeded Time complexity: O(n^2) Space complexity: O(1)

```java
// Java
class Solution {
    public boolean containsDuplicate(int[] nums) {
        int length = nums.length;
        if (length == 1) {
            return false;
        }
        for(int i=0;i<length;i++){
            int temp = nums[i];
            for(int j=0;j<i;j++){
                if(temp == nums[j]) {
                    return true;
                }
            }
        }
        return false;
    }
}
```

2. Sorting and then checking for duplicates Time complexity: O(nlogn) Space complexity: O(1)

```java
class Solution {
    public boolean containsDuplicate(int[] nums) {
        Arrays.sort(nums);
        int length = nums.length;
        for (int i=0;i<length-1;i++){
            if(nums[i]==nums[i+1]){
                return true;
            }
        }
        return false;
    }
}
```

3. Using HashSet Time complexity: O(n) Space complexity: O(n)

```java
class Solution {
    public boolean containsDuplicate(int[] nums) {
        HashSet<Integer> seen = new HashSet<>;
        for(int num: nums){
            if (seen.contains(num)){
                return true;
            }
            seen.add(num);
        }
        return false;
    }
}
```