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

Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

**Example 1:**

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

**Output:** true

**Example 2:**

**Input:** nums = [1,2,3,4]

**Output:** false

**Example 3:**

**Input:** nums = [1,1,1,3,3,4,3,2,4,2]

**Output:** true

**Constraints:**

1 <= nums.length <= 10^5

-10^9 <= nums[i] <= 10^9

**Attempt 1: 2024-06-15**

**Solution 1: Set (10 min)**

class Solution {

    public boolean containsDuplicate(int[] numbers) {

        // Initialize a HashSet to store unique elements.

        Set<Integer> uniqueNumbers = new HashSet<>();

        // Iterate through all the elements in the array.

        for (int number : numbers) {

            // Attempt to add the current element to the set.

            // If the add method returns false, it means the element is already present in the set.

            if (!uniqueNumbers.add(number)) {

                // Duplicate found, so return true.

                return true;

            }

        }

        // No duplicates were found, return false.

        return false;

    }

}

**Refer to**

<https://algo.monster/liteproblems/217>

**Time and Space Complexity**

The given Python code checks for duplicates in a list by converting it into a set, which stores only unique elements, and then comparing the length of this set to the length of the original list. If the set has fewer elements, it means that there were duplicates in the original list.

Time Complexity:

The main operation here is the conversion of the list into a set, which typically has a time complexity of O(n) where n is the number of elements in the list. This is because the operation needs to iterate through all elements once, adding them into the set and checking for uniqueness.

Space Complexity:

The space complexity is also O(n) in the worst case because in a situation where all elements are unique, the set would need to store all n elements separately from the original list.