**Program to find duplicate elements in an Array**

Approach 1) Brute-Force Method

In this method, we compare each element in the Array with every other element. 2 FOR loops are employed to achieve the same.

The complexity of this approach is O(n^2). The performance of this method is slow if there are lots of elements in the Array. So, this method is not recommended in real time.

Approach 2) Sorting the Array before Iteration

In this method, the Array is sorted first. This will put the identical elements next to each other.

Once the array is sorted, we can simply iterate through the array and check the previous element in the array and if it equals the current element, then we have a duplicate.

The complexity of this approach depends upon the complexity of the algorithm used for sorting. Bubble sort, Insertion sort and Selection sort have O(n^2). Merge sort and Heap sort have somewhat better complexity of O(n log n).

Approach 3) Using HashSet

HashSet contains only unique elements. HashSet never allows duplicate elements. We use this property of HashSet to find duplicates in an array. We try to add each element of an array into HashSet using add() method. This method will return true if an element is added successfully otherwise it returns false.

We can also use HashMap and store the elements in the array as keys and number of occurrences as the corresponding value in the map.

The time complexity of add() or put() operation in HashSet and HashMap respectively is O(1). This complexity depends upon the implementation of Hash function.