Analysis

This problem is as straight-forward as its description is ⍰. We will be given a sorted array containing duplicates, and we need to keep only single instance of each element. Also, we need to update the array in-place which means we have to do all the manipulations in the same array without using any data structure.

At the end, we have to return the length of the updated array.

Approach

One important thing in the question is that the array is sorted. This means that all the duplicate elements will be adjacent to each other. For e.g., in the array [1,2,2,3,4,4,4,5,5,6,7], we see all the duplicate elements are adjacent to each other.

We can use this property of the sorted array containing duplicates to solve this question using the following steps —

1. Check if the current and the next element are equal.
2. If they are, we will skip the next element and will continue the procedure as long as we encounter the duplicates.
3. If the elements are not duplicate, we will place the next (different) element next to the current element.

Time Complexity

We are scanning the array once, hence the time complexity will be O(n).

Space Complexity

Since we are forbidden to use the external data structure (and we are not using it ⍰), the space complexity will be O(1).