# Removing Duplicated Arrays

```
class Solution:
    def removeDuplicates(self, nums: List[int]) -> int:
        if not nums:
            return False

        write_index = 1

        for i in range(1, len(nums)):
            if nums[i-1] != nums[i]:
                nums[write_index] = nums[i]
                write_index += 1
        return write_index
```

So at first I thought I had to turn it into sets to remove any duplicate elements and simply just return the length of the set, however the code asked me to modify the list as I count the number of unique elements

This code has a counter in which counts the unique elements as well as the position of the index, so its actually quite space efficient as it only deals with 1 array

the pointer starts at the 2nd element, and checks if the previous element is unique or not, if it is unique then the i = the to current position of the write_index, otherwise if it was the same number, it would carry on until it finds a unique number

So the loop invariant is in which elements from 0 to write index should be sorted numbers and contain a sequence of unique numbers