

[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



 *Search Course*

Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)

Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)

Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)

Problem Challenge 3
(/courses/grokking-the-coding-interview/...

Remove Duplicates (easy)

We'll cover the following ^

- Problem Statement
- Try it yourself
- Solution
 - Code
 - Time Complexity
 - Space Complexity
- Similar Questions

Problem Statement

Given an array of sorted numbers, **remove all duplicates** from it. You should **not use any extra space**; after removing the duplicates in-place return the length of the subarray that has no duplicate in it.

Example 1:

[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



- Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)
- Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)
- Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)
- Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vxrMq3)
- Problem Challenge 3
(/courses/grokking-the-coding-interview/...

Input: [2, 3, 3, 3, 6, 9, 9]

Output: 4

Explanation: The first four elements after removing the duplicates will be [2, 3, 6, 9].



Example 2:

Input: [2, 2, 2, 11]

Output: 2

Explanation: The first two elements after removing the duplicates will be [2, 11].

Try it yourself

Try solving this question here:

Java

Python3

JS

C++


```
1 class RemoveDuplicates {
2
3     public static int remove(int[] arr) {
4         // TODO: Write your code here
5         int ans = 1;
6         for(int i =1 ;i<arr.length;i++)
7         {
8             if(arr[i]!=arr[i-1])
9             {
10                ans++;
11            }
12        }
13        return ans;
14    }
15 }
16
```



[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed 



- Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)
- Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)
- Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)
- Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- Problem Challenge 3
(/courses/grokking-the-coding-interview/...

⚙️📄

TestSave *Reset🔍

Show ResultsShow Console

📋 2 of 2 Tests Passed

Result	Input	Expected Output	Actual Output	Reason
✓	remove([2, 3, 3, 3, 6, 9, 9])	4	4	Succeeded
✓	remove([2, 2, 2, 11])	2	2	Succeeded

3.009s


Solution

In this problem, we need to remove the duplicates in-place such that the resultant length of the array remains sorted. As the input array is sorted, therefore, one way to do this is to shift the elements left whenever we encounter duplicates. In other words, we will keep one pointer for iterating the array and one pointer for placing the next non-duplicate number. So our algorithm will be to iterate the array and whenever we see a non-duplicate number we move it next to the last non-duplicate number we've seen.

Here is the visual representation of this algorithm for Example-1:

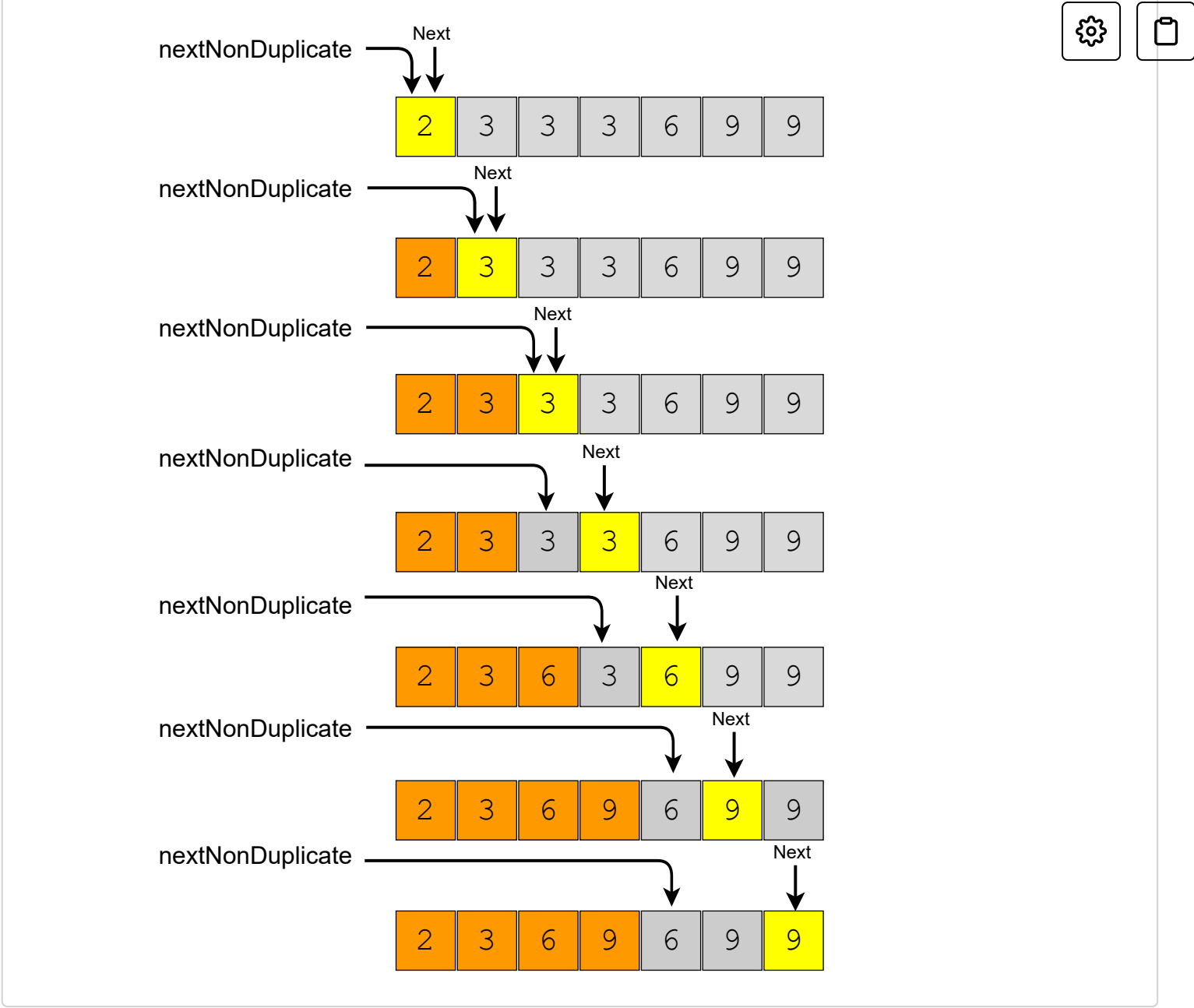
[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed 



Problem Challenge 1	(/courses/grokking-the-coding-interview/N8vB7OVYo2D)
Solution Review: Problem Challenge 1	(/courses/grokking-the-coding-interview/N0o9QnPLKNv)
Problem Challenge 2	(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)
Solution Review: Problem Challenge 2	(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
Problem Challenge 3	(/courses/grokking-the-coding-interview/...



Code #

Here is what our algorithm will look like:



[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



- Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)
- Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)
- Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)
- Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- Problem Challenge 3
(/courses/grokking-the-coding-interview/...

Java Python3 C++ JS

```
1 class RemoveDuplicates {
2
3     public static int remove(int[] arr) {
4         int nextNonDuplicate = 1; // index of the next
5         for (int i = 1; i < arr.length; i++) {
6             if (arr[nextNonDuplicate - 1] != arr[i]) {
7                 arr[nextNonDuplicate] = arr[i];
8                 nextNonDuplicate++;
9             }
10        }
11
12        return nextNonDuplicate;
13    }
14
15    public static void main(String[] args) {
16        int[] arr = new int[] { 2, 3, 3, 3, 6, 9, 9 };
17        System.out.println(RemoveDuplicates.remove(arr));
18
19        arr = new int[] { 2, 2, 2, 11 };
20        System.out.println(RemoveDuplicates.remove(arr));
21    }
22 }
23
```

Run

Save

Reset



Time Complexity

The time complexity of the above algorithm will be $O(N)$, where 'N' is the total number of elements in the given array.

Space Complexity

The algorithm runs in constant space $O(1)$.



[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)

Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)

Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)

Problem Challenge 3
(/courses/grokking-the-coding-interview/...

Similar Questions

Problem 1: Given an unsorted array of numbers and a target 'key', remove all instances of 'key' in-place and return the new length of the array.

Example 1:

Input: [3, 2, 3, 6, 3, 10, 9, 3], Key=3

Output: 4

Explanation: The first four elements after removing every 'Key' will be [2, 6, 10, 9].

Example 2:

Input: [2, 11, 2, 2, 1], Key=2

Output: 2

Explanation: The first two elements after removing every 'Key' will be [11, 1].

Solution: This problem is quite similar to our parent problem. We can follow a two-pointer approach and shift numbers left upon encountering the 'key'. Here is what the code will look like:



Java



Python3



C++



JS

[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



Problem Challenge 1
(/courses/grokking-the-coding-interview/N8vB7OVYo2D)

Solution Review: Problem Challenge 1
(/courses/grokking-the-coding-interview/N0o9QnPLKNv)

Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)

Problem Challenge 3
(/courses/grokking-the-coding-interview/...

```
class RemoveElement {

    public static int remove(int[] arr, int key) {
        int nextElement = 0; // index of the next element which is not 'key'
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] != key) {
                arr[nextElement] = arr[i];
                nextElement++;
            }
        }

        return nextElement;
    }

    public static void main(String[] args) {
        int[] arr = new int[] { 3, 2, 3, 6, 3, 10, 9, 3 };
        System.out.println(RemoveElement.remove(arr, 3));

        arr = new int[] { 2, 11, 2, 2, 1 };
        System.out.println(RemoveElement.remove(arr, 2));
    }
}
```

Run

Save

Reset



Time and Space Complexity: The time complexity of the above algorithm will be $O(N)$, where 'N' is the total number of elements in the given array.

The algorithm runs in constant space $O(1)$.

[← Back](#)

Pair with Target Sum (easy)

[Next →](#)

Squaring a Sorted Array (easy)

☒ Mark as Completed



[← Back To Course Home](#)

Grokking the Coding Interview: Patterns for Coding Questions

6% completed



Report an Issue



Ask a Question

(https://discuss.educative.io/tag/remove-duplicates-easy__pattern-two-pointers__grokking-the-coding-interview-patterns-for-coding-questions)



Search Course



Problem Challenge 1

(/courses/grokking-the-coding-interview/N8vB7OVYo2D)



Solution Review: Problem Challenge 1

(/courses/grokking-the-coding-interview/N0o9QnPLKNv)



Problem Challenge 2

(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)



Solution Review: Problem Challenge 2

(/courses/grokking-the-coding-interview/xl2g3vvrMq3)



Problem Challenge 3

(/courses/grokking-the-coding-interview/...