LeetCode Solutions

Sankalp Sangle

Last updated on May 2, 2020

Contents

Part I

LeetCode Top Interview Questions

Chapter 1

Easy

Link: LeetCode Top Interview Questions: Easy section.

1.1 Arrays

Link: Arrays

1.1.1 26. Remove Duplicates from Sorted Array

Link to question, Link to submission

Concepts Two pointer

Algorithm description

- Maintain a read pointer and a write pointer, both starting from zero.
- Advance the write pointer until you see a new value or reach end of array.
- Write value at write location into read location.

- Return read.

1.1.2 122. Best Time to Buy and Sell Stock II

Link to question, Link to submission

Concepts Greedy

Algorithm description

- Construct a consecutive elements difference array
- Return sum of all positive elements in difference array

1.1.3 189. Rotate Array

Link to question, Link to submission approach 1, Link to submission approach 2

Concepts Cyclic replacements, Implementation

Approach 1 description

- Maintain a visited array and a pointer initialized to 0
- while pointer + k is not visited, replace arr[pointer + k] with arr[pointer]. Update pointer to pointer + k. Set pointer + k to visited, increment a numberOfChanges variable.
- Increment pointer by 1
- Keep doing this while numberOfChanges less than size of array.

Approach 2 description

- Reverse the entire array
- Reverse from start to start + k
- Reverse from start + k to end

1.1.4 217. Contains Duplicate

Link to question, Link to submission

Concepts Hash Table, Set

Algorithm description

- Initialize a Set
- For an element in array, if element in Set, return true
- else add element to Set
- If out of loop, return False

1.1.5 136. Single Number

Link to question, Link to submission

Concepts Bit Manipulation, XOR

Algorithm description

- Initialize an answer variable to 0
- For every element, XOR it to answer. Elements appearing twice get XOR'd out to zero

- Return answer

1.1.6 350. Intersection of Two Arrays II

Link to question, Link to submission approach 1, Link to submission approach 2

Concepts Hash Table, Two Pointers

Approach 1 description

- Form an element:frequency mapping using map for smaller array (to save space)
- Traverse bigger array
- If frequency of element less than 0, add to answer. Decrement frequency

Approach 2 description

- If arrays are sorted, use two pointers p1 and p2
- If nums1[p1] == nums2[p2], add to answer and increment both
- Else if nums1[p1] is smaller, increment p1. Else increment p2
- Keep doing until reach end of either array

1.1.7 66. Plus One

Link to question, Link to submission

Concepts Array

Algorithm description

- Initialize a carry variable to 1
- Traverse array from the end.
  digit[i] = carry + digit mod 10, carry = carry + digit div 10
- Finally, if carry is not zero, insert carry at start of array

1.1.8 283. Move Zeroes

Link to question, Link to submission

Concepts Two Pointers

Algorithm description

- Maintain a read and a write pointer, both initialized to 0
- if read end has zero, increment read end
- else, copy read end to write end and increment both
- After read end reaches end, set all numbers from write end to end as 0

1.1.9 1. Two Sum

Link to question, Link to submission approach 1, Link to submission approach 2

Concepts Hash Table, Two Pointer

Approach 1 description

- Create an element:indices mapping
- Sort the array
- Use two pointers to search for a particular sum
- Once you find the sum, pop index from left pointer, and pop index from right pointer
- Return indices

Approach 2 description

- Create a hashmap of int, int
- Iterate the array with i as looping variable
- If element in hashmap, return (hashmap[element], i)
- Else insert hashmap[target - element] = i

1.1.10 36. Valid Sudoku

Link to question, Link to submission

Concepts Hash Table, Set

Algorithm description

- Create sets to hold numbers for each row, col and square.
- Traverse the sudoku
- If a number is already in the row, col, square, return False
- Else, come out of loop and return true

1.1.11 48. Rotate Image

Link to question, Link to submission

Concepts Array, Circular Permutation

Algorithm description

- Do a counterclockwise circular permutation as mentioned in solution
- Pure implementation problem. No algorithmic skill.

1.2 Strings

Link: Strings

1.2.1 344. Reverse String

Link to question, Link to submission

Concepts Two Pointers

Algorithm description

-