Random Leetcode Problems

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Level | Leetcode Problem | On Local and GitHub | Notebook | Last Visited | Status |
|  | Easy | Roman to Integer | Yes | Yes | 08/15/2022 | ✔️ -Make a map of only the values given in problem. -The keys are letter strings.  -If s[i+1]>s[i], subtract s[i]. If not, add s[i]. |
|  | Easy | Make Array Zero by Subtracting Equal Amounts | Yes | Yes | 09/12/2022 | -Hash map problem.-Answer is = number of unique elements that are not zero |
|  | Easy | Word Pattern | Yes | Yes | 09/12/2022 | -Hash map problem |
|  | Easy | Palindrome  Number |  |  |  | FINISH |
|  | Easy | Palindrome Permutation | Yes | Yes | 09/12/2022 | -Use an object. Traverse over array and if we haven’t seen character before, add it to object with freq of 1.  -If we have seen character before, delete one freq of character.  -Traverse over objectkeys, return true if zero or one character that occur odd number of times |
|  | Easy | Longest Common Prefix | Yes | Yes | 07/07/2022 | FINISH |
|  | Easy | Valid Parentheses | Yes | Yes | 09/18/2022 |  |
|  | Easy | Remove Duplicates from Sorted Array | Yes | Yes | 07/11/2022 | ✔️ |
|  | Easy | Remove Element | Yes | Yes | 08/15/2022 | ✔️ This is a two pointers problem |
|  | Easy | First Unique Character in a String | No | Yes | 08/15/2022 | -Must iterate through array twice, but only O(n) time. |
|  | Easy | [Maximum Difference Between Increasing Elements](https://leetcode.com/problems/maximum-difference-between-increasing-elements) | No | No | 08/16/2022 | ✔️Solved just like Best Time to Buy and Sell Stock |
|  | Easy | Rotate String | Yes | Yes | 08/18/2022 | ✔️Add string to string to get double of input string. Search the double for goal string using the includes method. |
|  | Easy | Check if Array Is Sorted and Rotated | Yes | Yes | 08/23/2022 | Check for more than one break point arr[i]>arr[i+1]. If so, false. Also check for special case when arr[0]>arr[len-1] |
|  | Easy | Remove Duplicates from Sorted List | No | Yes | 09/16/2022 | -Linked List problem-While loop should be (while current)-inside while loop if statement should say (if current.next =! Null and current.next=current.next.next)-inside while loop else statement should make current=current.next |
|  | Easy | Reverse String | Yes | Yes | 09/19/2022 | -while (left <=right) |
|  | Easy | Move Zeros | No | Yes | 08/24/2022 | Revisit |
|  | Easy | Intersection of Two Arrays | No | Yes | 08/26/2022 | -Create two sets. One set includes all values (non-unique) in arr1, other set has all non-unique values of arr2.Iterate over one set and check for values that are also present in set 2. If present in both, add to new array |
|  | Easy | Single Number | Yes | Yes | 09/06/2022 | - FINISH when you learn bit manipulation-For now, sort the array. Then check if arr[i]=arr[i+1], if not return i |
|  | Easy | Find the Difference | Yes | Yes | 08/27/2022 | -Use map and iterate over first string. Add each character to map as key and value should be each time character appears in string.-Iterate over string t and decrease map value for each key in string t-Any letter in t that is not in map is answer |
|  |  | Merge Two Sorted Lists | Yes | Yes | 07/11/2022 | Revisit |
|  |  | Implement strStr() | Yes | Yes | 07/11/2022 | -Verify Big O Accuracy of Time and Space Complexity of Approach 2 |
|  |  | Plus One | Yes | Yes | 08/07/2022 | -Verify Big O Space Complexity |
|  |  | Sqrt(x) |  |  |  | FINISH |
|  |  | Contains Duplicates II (Sliding Window) | Yes | Yes | 07/26/2022 | ✔️ |
|  |  | How Many Numbers Are Smaller Than Current Number |  |  | 07/26/2022 | -See if I can find more time optimal solution |
|  |  | Missing Number | Yes | Yes | 08/26/2022 | -sum=0 before loop. For loop iterating over nums. For every element in loop, sum += i +1 -nums[i], return sum |
|  |  | Minimum Size Subarray Sum | Yes | Yes | 07/28/2022 | ✔️ |
|  |  | Maximum Subarray of Size K | Yes | Yes | 07/28/2022 | -Revisit |
|  | Medium | Subarray Sum Equals K | Yes | Yes | 09/21/2022 | -Revisit-Create variable count=0, sum=0-Then, create a map. First map entry is 0: 1. We have seen sum of 0 one time so far, as cumulative sum is initially 0.-Iterate over nums. For each element in nums, cum sum= cum sum + nums[i]- On same iteration, If the map has (cum sum-k), increase count and then increase the amount of times we have seen the amount cum sum-k in the map-If map does not have cumulative sum at nums[i], add to map with count of 1-If map has the sum itself, increase map.get(sum) by value of 1 |
|  | Medium | Search a 2D Matrix | Yes | Yes | 09/09/2022 | -Binary Search on entire array |
|  |  | How Many Numbers Are Smaller Than the Current Number |  |  | 07/28/2022 | Revisit |
|  |  | 1D Array into 2D Array | Yes | Yes | 08/01/2022 | Figure out space complexity |
|  |  | Logger Rate Limiter | No, not necessary | Yes | 08/01/2022 | ✔️ |
|  |  | Minimum Area Rectangle | Yes | Yes | 08/04/2022 | ✔️, Time Complexity is O(n^2) regardless of how you solve it |
|  |  | Minimum Number of Operations to Move All Balls to Each Box | Yes | Yes | 08/04/2022 | Revisit |
|  |  | Meeting Rooms | Yes | Yes | 08/08/2022 |  |
|  |  | DI String Match | Yes | Yes | 08/08/2022 | -Use two pointers, max and min. Set min to 0, set max to s.length. At the end push the min to new array and return new array |
|  |  | Shuffle String | Yes | Yes | 08/08/2022 |  |
|  | Easy | Majority Element | No | Yes | 08/26/2022 | -Sort then return value at Math.floor((nums.length-1)/2) |
|  |  | Merge Intervals | No | Yes | 08/10/2022 | ✔️ |
|  | Medium | Validate IP Address | No | Yes | 08/23/2022 | ✔️Need regex values |
|  | Medium | Expressive Words | No | Yes | 08/23/2022 | Revisit, find time complexity and space complexity |
|  | Medium | Rotate Array | Yes | Yes | 08/26/2022 | Revisit-k=k%nums.length-Reverse the array in its entirety first. Then reverse the array from 0, k-1. Then reverse array from k, end-Understand what makes the time complexity O(n) |
|  | Medium | Sort Colors (Dutch National Flag) | Yes | Yes | 08/26/2022 | Revisit |
|  | Medium | Maximum Subarray | Yes | Yes | 09/23/2022 | -Revisit-Kadane’s algorithm-var maxSum= nums[0], var currentSum= maxSum-for loop starting with i=1-At every iteration of i, currentSum= Math.max(nums[i], currentSum + nums[i]), max=Math.max(max, currentSum) |