Random Leetcode Problems

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Level | Leetcode Problem | On Local and GitHub | Notebook | Last Visited | Status |
|  |  | 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 |
|  |  | Minimum Size Subarray Sum | Yes | Yes | 07/28/2022 | ✔️ |
|  |  | Maximum Subarray of Size K | Yes | Yes | 07/28/2022 | -Revisit |
|  |  | How Many Numbers Are Smaller Than the Current Number |  |  | 07/28/2022 |  |
|  |  | 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 |  |
|  |  | Merge Intervals | No | Yes | 08/10/2022 | ✔️ |
|  | Easy | Roman to Integer | Yes | Yes | Revisited  11/16/2022 | -Make a map of only the values given in problem. -The keys are roman numeral strings.  -If s[i+1]>s[i], subtract s[i]. If not, add s[i]. |
|  | Easy | Find Pivot Index | Yes | Yes | Revisited  12/03/2022 | ✔️-Find total array sum. Then, traverse the array starting from the left while adding to left sum with every value. pivot index is where total sum- left sum-array [pivot index]=left sum |
|  | Easy | Contains Duplicate | Yes | Yes | 01/11/2023 | ✔️-Use sorting then array traversal or hash map/hash set. Most time and space optimal is sorting |
|  | Easy | Ugly Number | Yes | Yes | 11/17/2022 | -Revisit-To check if a number is ugly, we keep dividing the number by 2,3 or 5 until it becomes 1.-If the number cannot become 1 (no longer or never divisible by 2,3,5), then return false. |
|  | Easy | Remove All Adjacent Duplicates In String | Yes | Yes | 11/09/2022 | -Revisit-Create an empty array res. Traverse through string. If s[i]===res.pop(), remove the last element from res(pop). If not, res.push(s[i]). Return res.join() |
|  | Easy | Search Index Position | Yes | Yes | 10/11/2022 | -Revisit-Binary search problem-Why do we return left + 1? |
|  | Easy | Find Middle Index | Yes | No | Revisited  12/03/2022 | ✔️-Same approach/solution as Find Pivot Index^ |
|  | Easy | Climbing Stairs | Yes | Yes | 12/15/2022 |  |
|  | Easy | Implement Queue Using Stack | Yes | Yes | 12/15/2022 | -Revisit-Initialize three variables. Var stackOne=[], varStackTwo=[], var this.front=null-For queue dequeue (FIFO), if stackTwo.length == 0, while (stack1.length != 0), stackTwo.push(stackOne.pop())Then, return stack2.pop()-For pushing element on to queue, if stackOne.length=0, this.front=item to push, else stackOne.push(x)-For peeking queue, if stackTwo.length==0, return this.front, else return this.stack2[this.stack2.length-1] |
|  | Easy | Middle of Linked List | Yes | Yes | Revisited  12/05/2022 | -Revisit-Create two variables, fast and slow. Initialize them both to the value of head.-create a while loop with the condition (fast & fast.next)-while fast & fast.next, fast=fast.next.next, slow=slow.next-return slow-When traversing the list with a pointer slow, make another pointer fast that traverses twice as fast. When fast reaches the end of the list, slow must be in the middle. |
|  | Easy | Make Array Zero by Subtracting Equal Amounts | Yes | Yes | 09/12/2022 | -Revisit-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 | -Use a “stack” array |
|  | 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 | Determine If String Halves Are Alike | Yes | Yes | Revisited  12/05/2022 | ✔️-Make a set with all the vowels in it (capitalized and lowercase) |
|  | 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 | Range Sum of BST | Yes | Yes | 12/10/2022 | Revisit-DFS: Iterative or Recursive-Create a stack variable which is an empty array-Create a sum variable and initialize it to 0-Add root to stack-while loop with the condition while(stack.length)-node=stack.pop()-if node.val is => low and <=high, add node.val to sum(else add 0 to sum)-if(node.left) stack.push(node.left)-if(node.right) stack.push(node.right)-At the end, return sum |
|  | 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 |
|  | Easy | Missing Number | Yes | Yes | Revisited  12/05/2022 | -sum=0 before loop. For loop iterating over nums. For every element in loop, sum += i +1 -nums[i], return sum |
|  | Easy | Make The String Great | Yes | Yes | Revisited  12/05/2022 |  |
|  | Easy | Linked List Cycle | Yes | Yes | Revisited  12/05/2022 | -Revisit  -Check to see if head is null, if it is, return false  -Initialize a slowPointer variable and assign its value to be head  -Initialize a fastPointer variable and assign it’s value to be head.next  -Create a while loop with the condition (fastPointer.next != null && fastPointer.next.next != null)  -if (fastPointer === slowPointer) return true, otherwise return false |
|  | Easy | Majority Element | No | Yes | 08/26/2022 | -Sort then return value at Math.floor((nums.length-1)/2) |
|  | 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 | Group Anagrams | Yes | Yes | 01/11/2022 |  |
|  | Medium | Minimum Average Difference | Yes | Yes | 12/04/2022 | -Revisit-Prefix sum problem. Right sum should be initialized to total sum of nums.-Create variable minimum and initialize it to Infinity-Create variable ans and do noinitialize it to any value-Left sum should be initialized to 0.-Iterate over nums, for each value of i, add nums[i] to leftSum and subtract nums[i] from rightSum.-if the current minimum avg difference is less than minimum, update ans to i and minimum to the current minimum avg difference-return ans |
| 1. ] | Medium | Best Time to Buy And Sell Stock II | Yes | Yes | 11/17/2022 | -Revisit-Create a variable “profit” and set it equal to 0.-Iterate over the prices array with a for loop in which I is initialized to be equal to 1-If prices[i-1]<prices[i], add (prices[i]-prices[i-1]) to profit |
|  | Medium | Search a 2D Matrix | Yes | Yes | 09/09/2022 | -Binary Search on entire array |
|  | Medium | Online Stock Span | Yes | Yes | 11/09/2022 | -Revisit-Most optimal answer uses monotonic stack. Set up an array in the stock Spanner class. This.stack=[]- |
|  | Medium | Longest Palindrome by Concatenating Two Letter Words | Yes | Yes | 11/03/2022 | -Revisit--Revisit, understand more optimal solution?-Make a variable called length that starts at 0-Make a hash map and for each word in array, check if the reverse of the word is in map. If it is, add 4 characters to length. Then decrease the count of the reversed by 1 in map.-If reversed word does not exist in map, add the word itself with a count of 1-After all the two letter words have been removed grab thefirst entry in map of two letter words that have same letter (for center of palindrome) and increase length by 2 |
|  | 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 | Revisited  12/02/2022 | -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) |
|  | Medium | Odd Even Linked List | Yes | Yes | 12/05/2022 | ✔️-Need 3 variables at start: var odd=head, var even=head.next,var evenHead=even-then create while loop that runs while (even != null && odd.next.next !=null) |