|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Neetcode Problem | Level | On Local and GitHub | Notebook | Last Visited | Status |
| Contains Duplicates | Easy | Yes | Yes | 01/11/2023 | ✔️-Use sorting then array traversal or hash map/hash set. Most time and space optimal is sorting |
| Valid Anagram |  | Yes | Yes | 01/19/2023 | ✔️ -Use a hash map or a counter array of fixed size 26 (to represent 26 lowercase letters, like in Group Anagrams). Loop over one string to count frequency of each character in string. Loop over second string and for each character in second string decrease frequency of that character in the fixed array or hash map. If the fixed array or hash map does not have that character, return false. Otherwise, return true. |
| Two Sum |  | Yes | Yes | 07/04/2022 | -Need to verify Big O space complexity of Approach 1 and 2 |
| Group Anagrams |  | Yes | Yes | 01/11/2023 | -Revisit-First, create an empty object called results.-Then, create a for loop that loops over every string in the input array. Inside this first for loop, create an empty array called count of size 26 and fill each 26 slots with zeroes. For each string in input array, another for loop should run that iterates over every character in the string. For each char in string, count[char.charCodeAt()-97]++. Then close inner for loop. Inside outer for loop (for each string in input array), then create a variable called key, which is =count.join(‘#’). If res[key] exists, push current string to res[key]. If res[key] doesn’t exist, add res[key]=[current string] we are iterating over. |
| Top K Frequent |  | Yes | Yes | 07/05/2022 | -Learn Heaps then add new approaches  to notebook and local/GH |
| Product of Array Except Self |  |  |  |  |  |
| [Valid](https://leetcode.com/problems/product-of-array-except-self/) Sodoku |  |  |  |  |  |
| Encode and Decode Strings |  | Yes |  |  |  |
| Longest Consecutive Sequence |  | Yes | Yes | 09/23/2022 | -create a set and add all of nums into set  -loop over nums. If nums [i]-1 is not in set, set count =0;  -if nums[i]-1 is not in set, start a while loop that goes while nums[i]+1 exists in set. For each additional increase of 1, increment count by 1 |

Two Pointers(Verify Big O Accuracy for All)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neetcode Problem | On Local and GitHub | Notebook | Last Visited | Status |
| Valid Palindrome | Yes | Yes | 09/11/2022 | ✔️ |
| Two Sum II |  |  | 07/26/2022 |  |
| 3 Sum |  |  | 07/26/2022 |  |
| Container with Most Water |  |  |  |  |
| Trapping Rain Water |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Sliding Window(See Leetcode Doc for More)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neetcode Problem | On Local and GitHub | Notebook | Last Visited | Status |
| Best Time to Buy and Sell Stock | Yes | Yes | 07/28/2022 | ✔️ |
| Longest Substring without Repeating Characters | Yes | Yes | 09/12/2022 | -Sliding window |
| Longest Repeating Character Placement |  |  |  |  |
| Permutation in String |  |  |  |  |
| Minimum Window Substring |  |  |  |  |
| Sliding Window Maximum |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Linked Lists

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Neetcode Problem | Level | On Local and GitHub | Notebook | Last Visited | Status |
| Reverse Linked List |  | Yes | Yes | 08/01/2022 | Three variables: prev, current, next |
| Merge Two Linked Lists |  |  |  |  |  |
| Add Two Numbers | Medium | Yes | Yes | 09/23/2022 | -Revisit  -Requires a few modulo expressions, look into that again |
| Find the Duplicate Number |  |  |  |  |  |
| Linked List Cycle |  | Yes | Yes | 08/01/2022 | Fast and slow pointers(2) is most space & time optimal. Can also solve using hash map. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |