

# **Leetcode Questions (Following Neetcode Tree)**

**Rodrigo Ferreira**

Dec 23 - Prob Never



# Table of Contents

Table of Contents.....	3
Figures.....	4
Chapter 1 – Introduction.....	5
Chapter 2 – Arrays and Hashing.....	6
2.1 – 217 Contains Duplicates.....	6
2.2 – 242 Valid Anagram.....	6
2.3 – 1 Two Sum.....	7
2.4 – 49 Group Anagrams.....	7
2.5 – 347 Top K Frequent Elements.....	7
2.6 – 238 Product of Array Except Self.....	7
2.7 – 36 Valid Sudoku.....	7
2.8 – 128 Longest Consecutive Sequence.....	7
Chapter 3 – Stacks.....	7
3.1 - 20 Valid Parentheses.....	7

## Figures

## Chapter 1 – Introduction

In the ever-evolving landscape of technology, where innovation drives progress and excellence are paramount, the mastery of data structures and algorithms stands as a foundational pillar for any aspiring programmer or computer scientist. It is within this realm of intricate problem-solving and algorithmic wizardry that the journey with LeetCode challenges begins—a testament to one's unwavering desire to unravel the complexities of computational logic and refine the art of efficient coding.

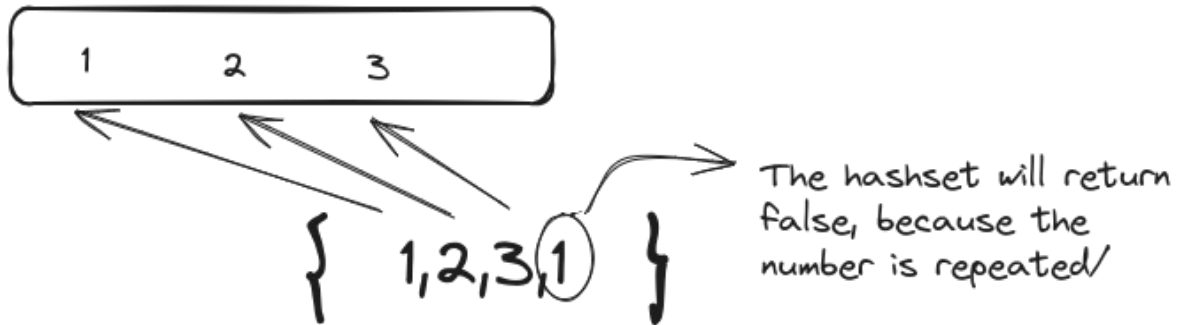
For many, including myself, the decision to embark on this challenging odyssey stems from a profound yearning to deepen our understanding and mastery of these fundamental concepts. As I delved into the world of computer science, fuelled by an insatiable curiosity and a relentless pursuit of excellence, the allure of LeetCode challenges beckoned as an unparalleled opportunity to put theory into practice and hone my skills in real-world problem-solving scenarios.

This paper serves as a testament to the transformative power of embarking on the journey with LeetCode challenges—a journey marked by trials and triumphs, setbacks and breakthroughs, yet ultimately defined by the inexorable pursuit of knowledge and mastery. Through a reflective exploration of my experiences and insights gained along this path, we aim to unravel the profound impact that LeetCode challenges have had on shaping not only my technical proficiency but also my mindset, resilience, and approach to tackling complex problems in the realm of computer science. Join me as we traverse the winding roads of this challenging journey, guided by the beacon of curiosity, determination, and the relentless pursuit of excellence.

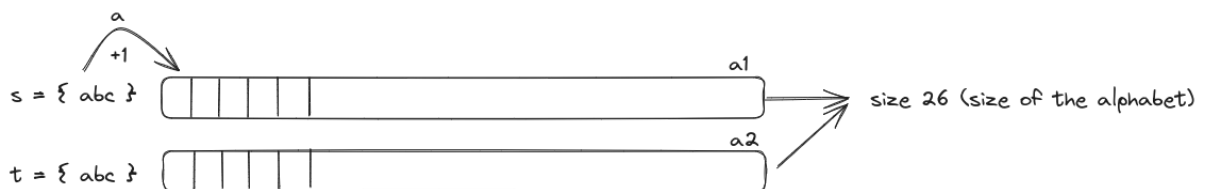
## Chapter 2 – Arrays and Hashing

### 2.1 – 217 Contains Duplicates

Hashset



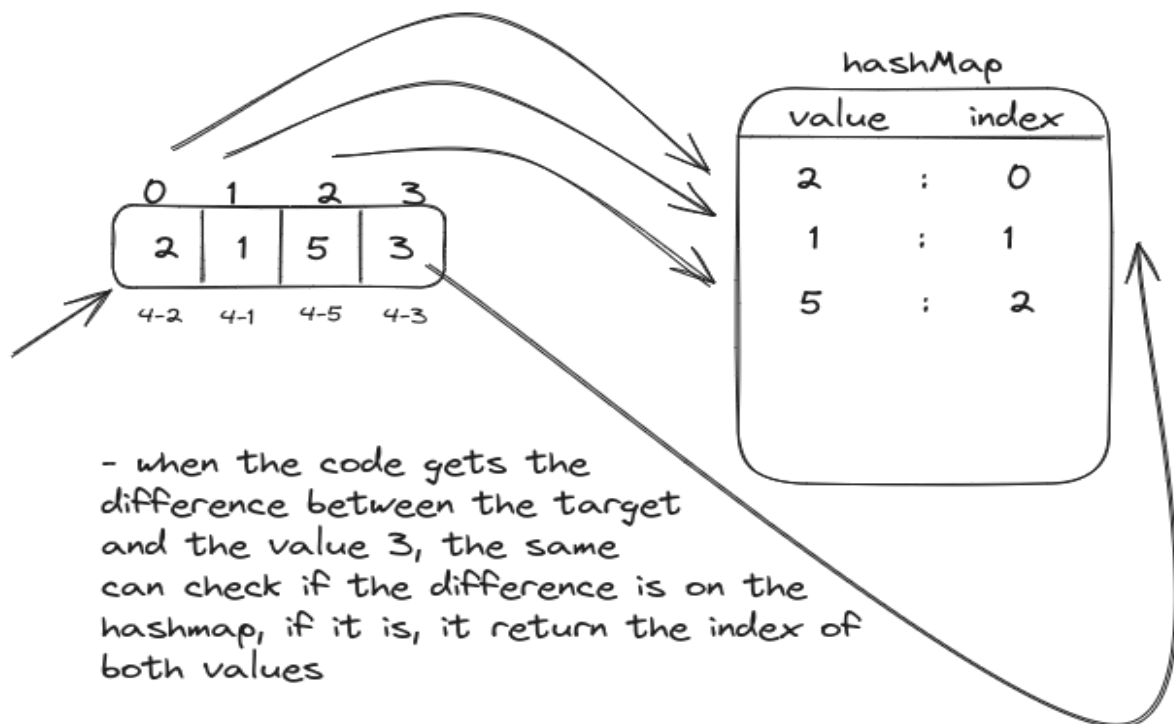
### 2.2 – 242 Valid Anagram



- the letter a will be the position 0 and z position 25
- after counting the letters, compare the a1 and a2, if they are the same, the words are a valid anagram

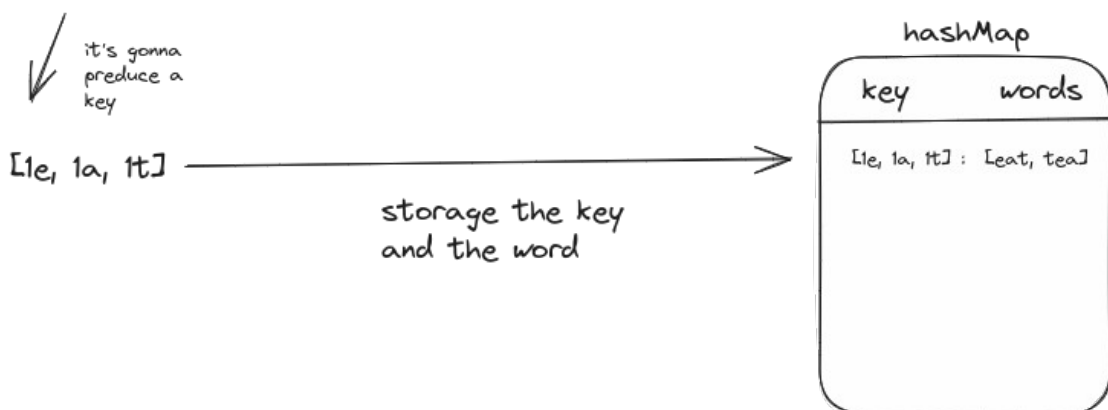
### 2.3 - 1 Two Sum

target = 4



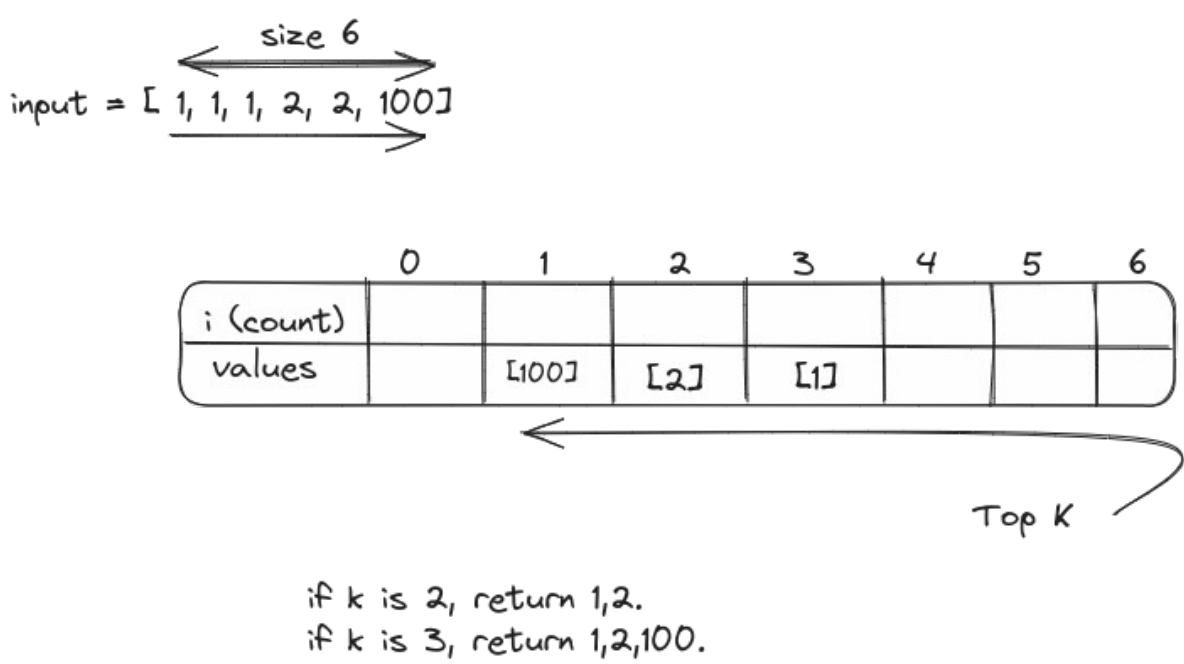
### 2.4 - 49 Group Anagrams

s = ["eat", "tea", "tan", "ate", "nat", "bat"]

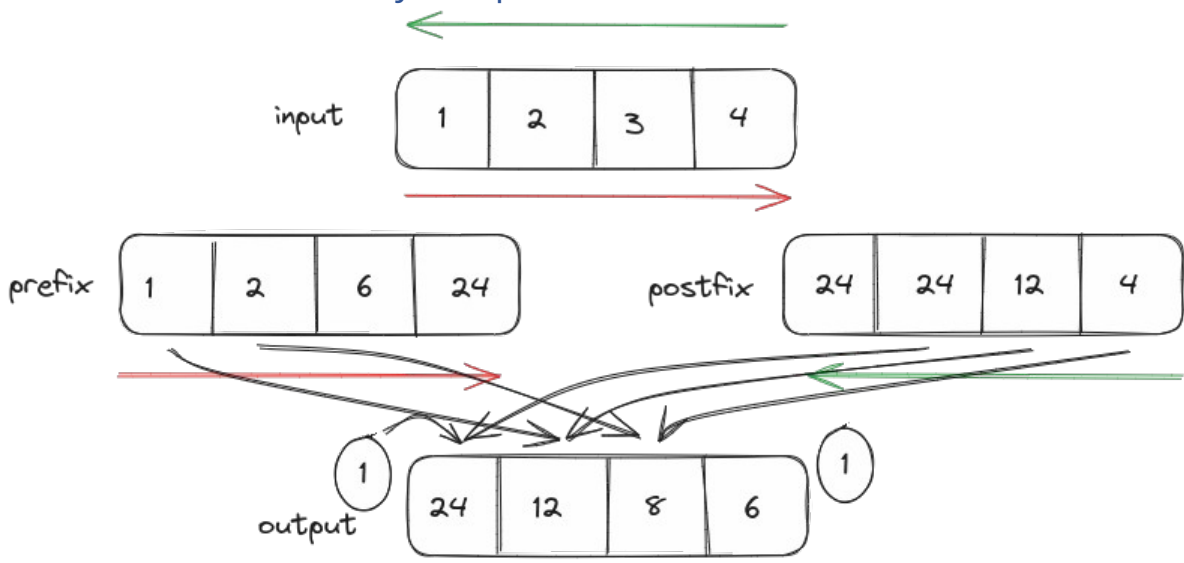


- do that for every word, and then print the organized words in the hashMap

### 2.5 – 347 Top K Frequent Elements



### 2.6 – 238 Product of Array Except Self



### 2.7 – 36 Valid Sudoku

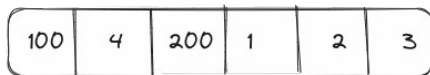
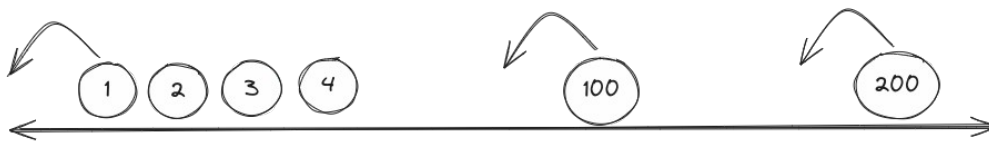
-> Create a dictionary for every row, col and block:

```
collections.defaultdic(set)
```

1. if the number is not there, add it.
2. otherwise, it's the sudoku is wrong.



## 2.8 - 128 Longest Consecutive Sequence



1. Pass the array to a set()
2. In each number, check if there in the set there is the previous number.
  - if there is a previous number, it means it's not the start of the set and we can drop it.
  - if there is not a previous number, it is the start of a set, and then we can check for the rest of the numbers.

100 ->  
 4 x  
 200 ->  
 1 -> 2 -> 3 -> 4  
 2 x  
 3 x

return 4, with  $O(n)$  time comp.

## Chapter 3 - Stacks

### 3.1 - 20 Valid Parentheses

