

Assignment

Arrays:

1 . Find the Second Largest Element in an Array

- Example: Input: `[1, 3, 4, 2]` → Output: `3`

2 . Move All Zeroes to the End

- Example: Input: `[0, 1, 0, 3, 12]` → Output: `[1, 3, 12, 0, 0]`

3 . Find the Union and Intersection of Two Arrays

- Example: Input: `arr1 = [1, 2, 3], arr2 = [2, 3, 4]`
Output: Union: `[1, 2, 3, 4]`, Intersection: `[2, 3]`

4. Rotate an Array by K Steps

- Example: Input: `arr = [1, 2, 3, 4, 5], k = 2` → Output: `[4, 5, 1, 2, 3]`

5. Merge Intervals

- Example: Input: `[[1, 3], [2, 6], [8, 10], [15, 18]]` → Output: `[[1, 6], [8, 10], [15, 18]]`

Strings

1 . Find the Frequency of Each Character

- Example: Input: `"apple"` → Output: `{ 'a': 1, 'p': 2, 'l': 1, 'e': 1 }`

2. Remove Duplicates from a String

- Example: Input: `"programming"` → Output: `"progamin"`

3. Find the First Non-Repeating Character

- Example: Input: `"swiss"` → Output: `"w"`

4 . Check if a String is a Rotation of Another

- Example: Input: `"abcde", "cdeab"` → Output: `True`

5. Compress a String Using the Counts of Repeated Characters

- Example: Input: "aaabbc" → Output: "a3b2c1"

Queues:

1. Implement a Queue Using Lists

- Implement enqueue, dequeue, and display operations using a Python list.

2 . Check for a Palindrome Using a Queue

- Example: Input: "radar" → Output: True

3 . Implement a Circular Queue

- Design and implement a circular queue with enqueue, dequeue, and display operations.

4 . Sort a Queue

- Sort the elements of a queue without using any extra data structure.
- Example: Input: [3, 1, 4, 2] → Output: [1, 2, 3, 4]

5 .Implement an LRU Cache

- Design a Least Recently Used (LRU) Cache using a queue and a hashmap.

Stack

1. Find the Minimum Element in a Stack

- Implement a special stack that supports push, pop, and retrieving the minimum element in O(1) time.

2. Reverse a String Using a Stack

- Example: Input: "hello" → Output: "olleh"

3 .Next Greater Element

- For each element in the array, find the next greater element.
- Example: Input: [4, 5, 2, 25] → Output: [5, 25, 25, -1]

4 . Implement a Stack Using Two Queues

- Design a stack using only two queues.

5 . Decode a String

- Decode strings with a pattern like "3[a2[c]]" → Output: "accaccacc"

Linked List

1 Detect a Loop in a Linked List

Example: Input: 1 -> 2 -> 3 -> 4 -> 2 (loop) → Output: True

2 . Remove Duplicates from a Sorted Linked List

- Example: Input: 1 -> 1 -> 2 -> 3 -> 3 -> None → Output: 1 -> 2 -> 3 -> None

3 . Check if a Linked List is a Palindrome

- Example: Input: 1 -> 2 -> 3 -> 2 -> 1 → Output: True

4 . Rotate a Linked List

- Example: Input: 1 -> 2 -> 3 -> 4 -> 5, k = 2 → Output: 4 -> 5 -> 1 -> 2 -> 3

5 .Add Two Numbers Represented by Linked Lists

- Example: Input: 7 -> 1 -> 6 (617) and 5 -> 9 -> 2 (295) → Output: 2 -> 1 -> 9 (912)