Linked Lists and Arrays

Task 1: -

* Read through the provided materials on arrays.

Arrays are basic data structures that store homogeneous elements(i.e, same type of elements) in contiguous memory locations .They are useful for storing and accessing data efficiently. Each element in an array is accessed using an index, which allows for quick retrieval and modification. Arrays are particularly handy when you need to work with a collection of items and perform operations like sorting or searching.

* Implement basic operations: creation, insertion, deletion, and traversal.

Link to the Code : <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/basicarrayoperations.java>

Here I have implemented above stated actions on arrays in multiple manners for each such as declaring and initialising in different ways and etc.,

* Solve practice problems related to arrays.

Solved practice problem questions given in the youtube link provided .

Question 1 : - Finding the Leaders of an Array. Leader is an element which is greater than all the elements present to the right side of itself.

Link to Code : <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/leadersofarray.java>

Time Complexity : -O(n) Space Complexity :-O(1)

Solved it using reverse traversal and a temporary leader variable.

Question 2:- Sort a Array which contains only 0,1 and 2s in O(n) time complexity

Link to Code : <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/sort012.java>

Solved it using pivots .

Time Complexity : -O(n) Space Complexity :-O(1)

Question 3:- Finding Smallest Missing Positive number in the given Array

Link to Code: <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/smallestmissingnumber.java>

Solved it using indexing using boolean array,

Time Complexity : -O(n) Space Complexity :-O(n)

Question 4:- Given a array , find all the elements which occur more than n/k times

Link to Code: <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/occurnces.java>

Solved this using a HashMap storing the occurrences and checking if the count is exceeding k at the same time

Time Complexity : -O(n) Space Complexity :-O(n)

Question 5:- Finding the Maximum sum of a subarray for a give array

Link to Code: <https://github.com/Praneethg23/PeopleTechIntern/blob/main/Arrays/maxsubarraysum.java>

Solved the problem using dynamic programming storing both current and maximum sums and updating them as required.

Time Complexity : -O(n) Space Complexity :-O(1)

Task 2:-

* Study the concept of linked lists.

Linked lists are a versatile data structure consisting of nodes, where each node contains a data element and a reference to the next node in the sequence. Unlike arrays, linked lists do not store elements in contiguous memory locations, which allows for efficient insertion and deletion operations without needing to shift elements. Linked lists come in various forms, such as singly linked lists, doubly linked lists, and circular linked lists. They provide themself very useful in data manipulation tasks.

* Implement basic operations: creation, insertion, deletion, and traversal.

Link To Code: <https://github.com/Praneethg23/PeopleTechIntern/tree/main/LinkedLists>

In the above link there is a concrete LinkedList class which have constructors to create linked list(single) and functions to create a node , insert a node at different positions such as start,end and specific position, delete a node at different positions , traversing through a list to find the length and also to print the elements in it. Additionally to these functions it also has a function to reverse all the links in the given list. There are also two testing codes where they test the Implemented Linked List Class.

* Compare and contrast arrays and linked lists in terms of performance and use cases.

Arrays and Linked Lists are both homogeneous and fundamental data structures but differ in implementation , performance ,use cases and etc.,

|  | Arrays | Linked Lists |
| --- | --- | --- |
| Access Time | Arrays are better in terms of accessing elements , as it is done by indexing with time complexity of O(1). | Linked lists have O(n) time complexity for accessing elements, as traversal is required. |
| Insertion/Deletion | Insertion and deletion is tougher in arrays as all the elements are to be shifted. | Here it is done much easier than arrays as it just deals with creating new link. |
| Memory Usage | Arrays definitely need contiguous blocks of memory and extending may not be possible in some cases | Linked lists use non-contiguous memory, reducing issues with memory allocation but increasing overhead due to storing pointers/references |
| Use cases | Better in static storage. Ideal for faster lookup and in heavy cache applications | Better for Dynamic Storage. Useful in systems where frequent insertions/deletions take place. |

Task 3: Hands On Practice

Solved questions from LeetCode

Linked List:

Question 1 :

Merge Two Sorted Lists. Difficulty Easy

Link : <https://leetcode.com/problems/merge-two-sorted-lists/description/>

Question 2:

Remove Duplicates from Sorted Lists. Difficulty Easy

<https://leetcode.com/problems/remove-duplicates-from-sorted-list/description/>

Question 3:

Add two numbers represented as Linked Lists. Difficulty Medium

<https://leetcode.com/problems/add-two-numbers/description/>

Question 4:

Remove Nth node from the end of the List .Difficulty Medium

<https://leetcode.com/problems/remove-nth-node-from-end-of-list/description/>

Question 5:

Swap Nodes in pairs .Difficulty Medium

<https://leetcode.com/problems/swap-nodes-in-pairs/description/>

Question 6:

Reverse Nodes in K-Groups .Difficulty Hard

<https://leetcode.com/problems/reverse-nodes-in-k-group/description/>

Question 7:

All O’one Data Structures. Difficulty Hard

<https://leetcode.com/problems/all-oone-data-structure/>

Solutions For the Above question are in the below Link

<https://github.com/Praneethg23/PeopleTechIntern/tree/main/LinkedLists/leetCode>

Arrays:

Question 1: Difficulty Easy

<https://leetcode.com/problems/two-sum/description/>

Question 2: Difficulty Easy

<https://leetcode.com/problems/search-insert-position/description/>

Question 3: Difficulty Medium

<https://leetcode.com/problems/container-with-most-water/description/>

Question 4: Difficulty Medium

<https://leetcode.com/problems/valid-sudoku/description/>

Question 5: Difficulty Hard

<https://leetcode.com/problems/median-of-two-sorted-arrays/description/>

Question 6: Difficulty Hard

<https://leetcode.com/problems/trapping-rain-water/description/>

Solutions for the above questions are in the below link

<https://github.com/Praneethg23/PeopleTechIntern/tree/main/Arrays/LeetCode>

All these questions I solved were using techniques such as dynamic programming and backtracking and also using Data Structures such as Heaps,Sets,HashMaps etc.