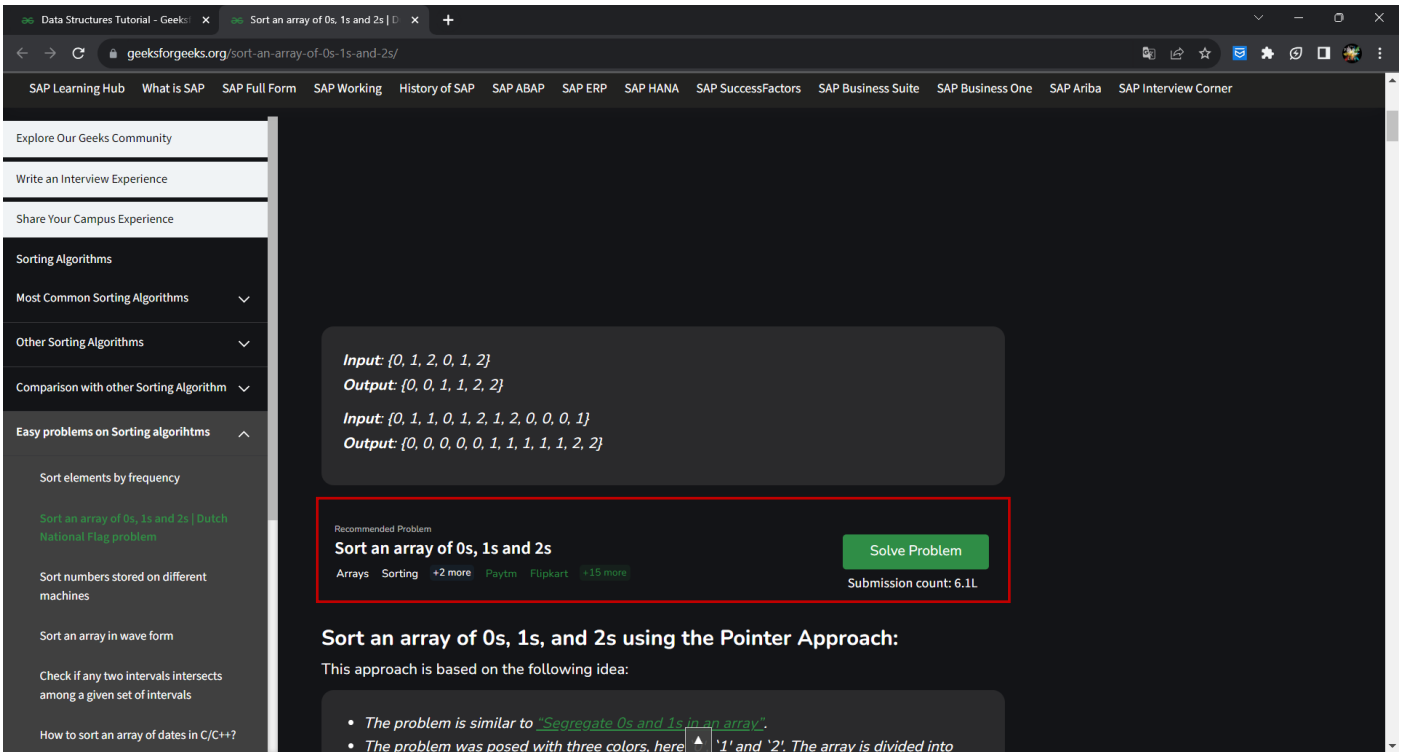
DATA STRUCTURES AND ALGORITHMS FURTHER PRACTISE

Tutorial: *Scrolling down and find Solve problem button*



I. Sorting:

- **Easy:**
  1. Sort elements by frequency
  2. Sort an array of 0s, 1s and 2s
  3. Sort numbers stored on different machines
  4. Sort an array in wave form
  5. Check if any two intervals overlap among a given set of intervals
  6. How to sort an array of dates in C/C++?
  7. Sorting Strings using Bubble Sort
  8. Find missing elements of a range
  9. Sort an array according to count of set bits
  10. Sort even-placed elements in increasing and odd-placed in decreasing order
  11. Sort an array when two halves are sorted
  12. Sorting Big Integers
  13. Sort a linked list of 0s, 1s and 2s
- **Medium:**
  1. Inversion count in Array using Merge Sort
  2. Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted
  3. Sort a nearly sorted (or K sorted) array
  4. Sort n numbers in range from 0 to n^2 – 1 in linear time
  5. Sort an array according to the order defined by another array
  6. Find the point where maximum intervals overlap
  7. Find a permutation that causes worst case of Merge Sort
  8. Sort Vector of Pairs in ascending order in C++
  9. Minimum swaps to make two arrays identical
  10. Chocolate Distribution Problem
  11. Permute two arrays such that sum of every pair is greater or equal to K
  12. Bucket Sort To Sort an Array with Negative Numbers
  13. Sort a Matrix in all way increasing order
  14. Convert an Array to reduced form using Vector of pairs
  15. Smallest Difference Triplet from Three arrays
  16. Check if it is possible to sort an array with conditional swapping of adjacent allowed
- **Hard:**
  1. Find Surpasser Count of each element in array
  2. Count distinct occurrences as a subsequence

II. Linked List:

- **Easy:**
    1. Print the middle of a given linked list
    2. Write a function that counts the number of times a given int occurs in a Linked List
    3. Check if a linked list is Circular Linked List
    4. Count nodes in Circular linked list
    5. Convert singly linked list into circular linked list
    6. Exchange first and last nodes in Circular Linked List
    7. Reverse a Doubly Linked List
    8. Program to find size of Doubly Linked List
    9. An interesting method to print reverse of a linked list
    10. Can we reverse a linked list in less than O(n)?
    11. Circular Linked List Traversal
    12. Delete a node in a Doubly Linked List
- **Medium:**
    1. Detect loop in a linked list
    2. Find length of loop in linked list
    3. Remove duplicates from a sorted linked list
    4. Intersection of two Sorted Linked Lists
    5. QuickSort on Singly Linked List
    6. Split a Circular Linked List into two halves
    7. Deletion from a Circular Linked List
    8. Merge Sort for Doubly Linked List
    9. Find pairs with given sum in doubly linked list
    10. Insert value in sorted way in a sorted doubly linked list
    11. Remove duplicates from an unsorted doubly linked list
    12. Rotate Doubly linked list by N nodes
    13. Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?
    14. Modify contents of Linked List
- **Hard:**
    1. Intersection point of two Linked Lists.

III. Stack:

- **Easy:**
    1. Infix to Postfix Conversion using Stack
    2. Prefix to Infix Conversion
    3. Prefix to Postfix Conversion
    4. Postfix to Prefix Conversion
    5. Postfix to Infix
    6. Convert Infix To Prefix Notation
    7. Check for balanced parentheses in an expression
    8. Arithmetic Expression Evalution
    9. Evaluation of Postfix Expression
    10. Reverse a stack using recursion
    11. Reverse individual words
    12. Reverse a string using stack
    13. Reversing a Queue
- **Medium:**
    1. How to create mergable stack?
    2. The Stock Span Problem
    3. Next Greater Element
    4. Next Greater Frequency Element
    5. Maximum product of indexes of next greater on left and right
    6. Iterative Tower of Hanoi
    7. Sort a stack using a temporary stack
    8. Reverse a stack without using extra space in O(n)
    9. Delete middle element of a stack
    10. Check if a queue can be sorted into another queue using a stack
    11. Check if an array is stack sortable
    12. Iterative Postorder Traversal | Set 1 (Using Two Stacks)
    13. Largest Rectangular Area in a Histogram | Set 2
    14. Find maximum of minimum for every window size in a given array
    15. Find index of closing bracket for a given opening bracket in an expression
    16. Find maximum difference between nearest left and right smaller elements
    17. Delete consecutive same words in a sequence
    18. Check mirror in n-ary tree
    19. Reverse a number using stack

20. Reversing the first K elements of a Queue

IV. Queue:

- **Easy:**
    1. Implement Stack using Queues
    2. Detect cycle in an undirected graph using BFS
    3. Breadth First Search or BFS for a Graph
    4. Traversing directory in Java using BFS
    5. Vertical order traversal of Binary Tree using Map
    6. Print Right View of a Binary Tree
    7. Find Minimum Depth of a Binary Tree
    8. Check whether a given graph is Bipartite or not
- **Medium:**
    1. Flatten a multilevel linked list
    2. Level with maximum number of nodes
    3. Find if there is a path between two vertices in a directed graph
    4. Print all nodes between two given levels in Binary Tree
    5. Find next right node of a given key
    6. Minimum steps to reach target by a Knight
    7. Islands in a graph using BFS
    8. Level order traversal line by line | Set 3 (Using One Queue)
    9. Find the first non-repeating character from a stream of characters

V. Tree:

Problems	Difficulty Level	Solve
Height of Binary Tree	Easy	Solve
Determine if two trees are identical	Easy	Solve
Mirror tree	Easy	Solve
Symmetric Tree	Easy	Solve
Diameter of tree	Easy	Solve
Checked for Balanced tree	Easy	Solve
Children Sum Parent	Easy	Solve
Check for BST	Easy	Solve
Array to BST	Easy	Solve
Largest value in each level of binary tree	Easy	Solve
Maximum GCD of siblings of a binary tree	Easy	Solve
Zigzag Tree Traversal	Easy	Solve
Inorder Successor in BST	Easy	Solve
Kth Largest Element in a BST	Easy	Solve
Check if subtree	Medium	Solve
Single Valued Subtree	Medium	Solve

Problems	Difficulty Level	Solve
Unique BSTs	Medium	Solve
Inorder Traversal (iterative)	Medium	Solve
Preorder Traversal (iterative)	Medium	Solve
Postorder Traversal(iterative)	Medium	Solve
Vertical Traversal of a Binary Tree	Medium	Solve
Boundary Traversal	Medium	Solve
Construct Binary Tree from Parent array	Medium	Solve
Construct Binary Tree from Preorder and Inorder Traversal	Medium	Solve
Preorder Traversal and BST	Medium	Solve
Construct tree from preorder traversal	Medium	Solve
Minimum distance between two given nodes	Medium	Solve
Maximum sum leaf to root path	Medium	Solve
Odd Even Level Difference	Medium	Solve
Lowest Common Ancestor of a Binary Tree	Medium	Solve
Ancestors in Binary Tree	Medium	Solve
Remove BST keys outside the given range	Medium	Solve
Pair with given target in BST	Medium	Solve

Problems	Difficulty Level	Solve
Sum Tree	Medium	Solve
BST to greater sum tree	Medium	Solve
BST to max heap	Medium	Solve
Clone binary tree with random pointer	Medium	Solve
Maximum sum of non adjacent nodes	Medium	Solve
Largest BST in a Binary Tree	Medium	Solve
Extreme nodes in alternate order	Medium	Solve
Connect nodes at same level	Hard	Solve
Nodes at given distance in a Binary Tree	Hard	Solve
Sorted Linked List to BST	Hard	Solve
Binary Tree to Doubly Linked List	Hard	Solve
Maximum sum path between two leaf nodes	Hard	Solve

VI. Heap:

- **Easy:**
    1. Heap Sort
    2. Check if a given Binary Tree is Heap
    3. How to check if a given array represents a Binary Heap?
    4. Iterative Heap Sort
    5. K'th Largest Element in an array
    6. K'th Smallest/Largest Element in Unsorted Array | Set 1
    7. Height of a complete binary tree (or Heap) with N nodes
    8. Heap Sort for decreasing order using min heap
- **Medium:**
    1. Sort an almost sorted array
    2. Print all nodes less than a value x in a Min Heap.
    3. Tournament Tree (Winner Tree) and Binary Heap
    4. Connect n ropes with minimum cost
    5. Maximum distinct elements after removing k elements
    6. K maximum sum combinations from two arrays
    7. Median of Stream of Running Integers using STL
    8. Median in a stream of integers (running integers)
    9. K'th largest element in a stream
    10. Largest triplet product in a stream
    11. Find k numbers with most occurrences in the given array
    12. Convert min Heap to max Heap
    13. Given level order traversal of a Binary Tree, check if the Tree is a Min-Heap

VII. Hashing:

Easy:

1. Find whether an array is subset of another array
2. Union and Intersection of two linked lists
3. Given an array A[] and a number x, check for pair in A[] with sum as x
4. Maximum distance between two occurrences of same element in array
5. Count maximum points on same line
6. Most frequent element in an array
7. Find the only repetitive element between 1 to n-1
8. How to check if two given sets are disjoint?
9. Non-overlapping sum of two sets
10. Check if two arrays are equal or not
11. Find missing elements of a range
12. Minimum number of subsets with distinct elements
13. Remove minimum number of elements such that no common element exist in both array
14. Find pairs with given sum such that elements of pair are in different rows
15. Count pairs with given sum
16. Count quadruples from four sorted arrays whose sum is equal to a given value x
17. Sort elements by frequency
18. Find all pairs (a, b) in an array such that a % b = k
19. Group words with same set of characters
20. k-th distinct (or non-repeating) element in an array.

Medium:

1. Find Itinerary from a given list of tickets
2. Find number of Employees Under every Employee
3. Longest subarray with sum divisible by k
4. Find the largest subarray with 0 sum
5. Longest Increasing consecutive subsequence
6. Count distinct elements in every window of size k
7. Design a data structure that supports insert, delete, search and getRandom in constant time

VIII. Graph:

Level 1

Problems	Solve
Print Adjacency List	Solve
BFS of Graph	Solve
DFS of Graph	Solve
Transitive Closure of a Graph	Solve
Union-Find	Solve
Detect Cycle using DSU	Solve

Level 2

Problems	Solve
Number of Provinces	Solve
Find the number of islands	Solve
Detect cycle in an undirected graph	Solve
Hamiltonian Path	Solve
Prerequisite Tasks	Solve
Course Schedule	Solve
Circle of Strings	Solve
Snake and Ladder problem	Solve
Bipartite Graph	Solve

Problems	Solve
Maximum Bipartite Matching	Solve
Detect cycle in a directed graph	Solve
Find whether path exists	Solve
Toplogical Sort	Solve
Level of Nodes	Solve
Possible paths between 2 vertices	Solve
X Total Shapes	Solve
Distance of nearest cell having 1	Solve
Mother Vertex	Solve
Unit Area of largest region of 1’s	Solve
Rotten Oranges	Solve
Minimum Swaps to Sort	Solve
Steps by Knight	Solve
Implementing Dijkstra Algorithm	Solve
Neeman’s Shoes	Solve
Minimum Spanning Tree	Solve

Problems	Solve
Strongly Connected Components (Kosaraju's Algo)	Solve
Bridge Edge in Graph	Solve
Flood Fill Algorithm	Solve
Replace O's with X's	Solve
Shortest Prime Path	Solve
Word Search	Solve
Construct binary palindrome by repeated appending and trimming	Solve
Word Boggle	Solve

Level 3

Problems	Solve
Critical Connections	Solve
Minimum Cost Path	Solve
Strongly Connected Components (Tarjan's Algo)	Solve