Syllabus overview

NOTE: Implementation | Library (Collection)

1. Searching & Sorting (Implementation | Library Functions) (2)
2. Recursion & Backtracking (2)
3. LinkedList (2)
4. Stack & Queues (2)
5. Binary Trees (2)
6. Binary Search Trees (1)
7. Heaps & Hashing (1)
8. Tries (Autocomplete) (1)
9. DP-1 DP-2 (3)
10. Greedy + Divide & Conquer (1)
11. Graph (DFS(Cycle Detection, Word Boggle) / BFS(Rotten Oranges, Word Boggle) / Topological Sorting (Alien Dictionary)) (1)
12. Projects (1-2) + Resume (1) + Mock Interview (1)

Here, (DMB2-P DONE) means the problem has been done in a practice session of DMB2 batch.

Searching & Sorting

Prerequisite

- Quicksort
- Mergesort
- Counting Sort / Bucket Sort
- External Sort

Searching

- Search an element in a sorted and rotated array (Medium)
- Find first and last positions of an element in a sorted array (Medium)
- Search in the sorted matrix (Medium)

Sorting

- Move Zeroes (Easy)
- Sort 0 1 2 (Medium)
- Alternative Sorting (Easy)
- Merge two sorted arrays (Easy)
- Merge k Sorted Lists - LeetCode (Hard)

Miscellaneous

- Find Majority Element (Medium)
- Find Peak Element In Array (Medium)
- Roman to Integer (Easy)
- Integer to Roman (Medium)
- Find the Missing Number (Easy)
- First Missing Positive (Hard)

- 8. (Hard)
-
- 9. Counting Inversions (Hard)(DMB2-P DONE)
-
- Container With Most Water (Medium) Trapping Rain Water (Hard)
-
- 11. Stock Buy Sell to Maximize Profit (Medium) (DMB2-P DONE)
-
- 12. Print a given matrix in the spiral form (Medium) (DMB1 DONE) (DMB2-P DONE)
-
- 13. Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1 (Hard)

- 14. Largest Sum Contiguous Subarray (Medium)
- 15. Minimum Number of Platforms Required for a Railway/Bus Station (Medium)(DMB2-P DONE)
- 16. (Easy)
- 17. (DMB1 DONE)(DMB2-P DONE)
- 18.
- 19. Create a dynamic array (Medium)
- 20. Longest Consecutive Sequence
- 21.
- 22.

H/W: Implement Searching/Sorting Algorithms - Binary Search, QuickSort, Merge Sort.

Read about External Sort.

Recursion & Backtracking

0. Fib Number and Factorial using tail recursion (Easy) (DMB2-I DONE)
1. Generate Parenthesis (Easy)
2. Print all possible combinations of the mobile keyboard (Hard)(DMB2-I DONE)
3. Count All possible Decoding (Medium)
4. Count Possible Paths (Easy to Medium)
5. Last Non-Zero Digit Factorial (Hard)
6. Write a program to print all permutations of a given string (Medium)
7. Rat in a maze (Medium)

8. Palindrome Partitioning (Hard)
9. Valid Sudoku (Hard)
10. Subset Sum | Backtracking-4 (Hard)
11. Combination Of Number (Medium)
12. N Queens Problem (Hard) /Print all configurations of it as H/W.
13. Generate IP Address (Hard)
14. Power subset (Medium)
15. H/W: Largest Number after k swaps
16. The Knight's tour problem | Backtracking-1 (Hard)
17. Leetcode #62 Unique Paths (Medium)
18. Excel Sheet Column Title (Easy)
19. 171. Excel Sheet Column Number (Easy)
20. Count trailing zeroes in factorial of a number (Easy)

LinkedList

0. Implement (Singly/Doubly/Circular) LinkedList class with CRUD operations
1. Reverse a linked list (Recursive /Iterative) (Easy)
2. Palindromic Linked List (Easy)
3. Add two numbers in the linked list (Medium)
4. Detect and remove a loop in the linked list (Medium)

5. Intersection point in the linked list (Easy to Medium)
6. Merge k Sorted Linked Lists (Easy to Medium)
7. Arrangement of Odd And Even Nodes In Linked List (Medium)
8. Remove all occurrences of duplicates from a sorted Linked List (Medium)
9. Merge two sorted linked lists (Easy)
10. Reverse Linked List in K groups (Medium)
11. Clone with Linked With Random Pointers (Medium to Hard)
12. Reorder Linked List (Hard)
13. Swap K nodes from the end (Medium to hard)
14. Reverse Alternate K nodes (Medium)
15. Sort a linked list (Hard)
16. Delete Node in linked List (Easy)
17. Length of longest Palindrome (Hard)
18. Function to check if a singly linked list is palindrome
19. Find-first-non-repeating-character-stream-characters (Hard)
20. Intersection of two Sorted Linked Lists
21. C/C++ Program for Remove duplicates from a sorted linked list
22. Remove duplicates from an unsorted linked list geeksforgeeks.org 196 Comments
23. C/C++ Program for Union and Intersection of two Linked Lists

H/W: Subtract Two Numbers, find the middle element in the linked list.

Stack & Queue

1. Implement Stack And Queues using Linked List, Arrays

2. Get min elements from the stack in constant time. (Medium to Hard)
3. Reverse the stack/queue (Medium)
4. Sort the stack/queue (Medium)
5. Implement Deque (Medium)
6. Sliding Window Maximum (Medium to Hard)
7. Largest Histogram (Hard)
8. Circular tour (Medium)
9. Balanced Parenthesis (Medium)
10. Celebrity Problem (Hard)
11. Merge Overlapping intervals (Medium to Hard)
12. Next Greater Element in Array (Medium)
13. Detect Duplicate Parenthesis (Easy to medium)
14. Nearest Smaller Number in Array (Medium),
15. Evaluate Expression (Hard)
16. 225. Implement Stack using Queues
17. Stack | Set 2 (Infix to Postfix)
18. LRU Cache
19. Minimum Window Substring

Trees

0. Implement traversals recursive - Inorder, Preorder, PostOrder, LevelOrder,

1. Implement traversals iterative - Inorder, Preorder, PostOrder, LevelOrder

2. Print Left/Right/Bottom/Top view of the Binary Tree

3. Construct tree from inorder and preorder traversal (Easy to Medium)

4. LCA of Binary Tree (Recursive/Iterative)

5. Diameter of Binary Tree

6. Sum of all nodes of Binary Tree (Easy)

7. Max Sum path from the leaf to leaf.

8. Mirror Tree / Identical tree (Easy)

9. Height of Binary Tree

10. Check if the tree is a (full binary tree/balanced binary tree/perfect binary tree) or not

11. Serialize/Deserialize Binary Tree

12. Connect Nodes on the same level (Hard)

13. Convert each level in Binary Tree to Doubly LinkedList (Hard)

14. Reverse Level Order, Spiral Level Order, Boundary Traversal, Vertical Traversal

15. Construct Special Binary Tree from given Inorder traversal

16. Print root to leaf path in Binary tree (Easy)

17. Print Cousins of a given Nodes in a binary tree

18. Print all nodes at K distance. (Hard)

19. Find Largest Subtree sum in Binary Tree (Easy to Medium)

H/W: Construct tree from inorder and postorder traversal (Easy to Medium)

20. Rotting Oranges

Binary Search Trees | Heap | Hashing | Disjoint Sets

0. Implement CRUD in BST (Medium)
a) INSERT
b) DELETE
c) SEARCH

1. Implement CRUD in Heap (Medium) => Priority Queue

2. H/W: Implement Heap Sort (Medium)

3. Construct BST from preorder traversal (East to Medium)

4. Median of a stream of running integers (Hard)

5. Merge K Sorted Arrays (Medium - Hard)

6. Kth Largest/Smallest Element in an array (Hard)

7. Largest BST in Binary Tree (Hard)

8. LCA of BST (Easy)

9. Inorder Successor in BST (Medium)

10. Sorted Array to BST (Easy)

11. Given n appointments, find all conflicting appointments (Hard) / Let's not talk about this as this question is the application of Interval Trees which is internally avl tree.

12. Find kth smallest element in BST (Order Statistics in BST) (Medium)

13. Construct BST from its given level order traversal (Hard)

14. Print BST keys in the given range (Easy)

15. Count of Smaller Numbers After Self (Hard)

16. Find Median from Data Stream (Heap)

- Talk about hashing algorithms
    - Implement Map (Ordered/Unordered Map)

Hashing

0. Implement Map / Collision Handling Techniques
1. Two Sum (Easy)
2. Length of the longest substring without repeating characters (Medium)
3. Find the smallest window in a string containing all characters of another string (Hard)
4. Design a data structure that supports insert, delete, search and getRandom in constant time (Medium to hard)
5. Tree Traversal such as vertical traversal, top, bottom, etc using Maps.
6.

Dynamic Programming

1. Coin Exchange Problem
2. Longest Common Substring
3. Longest Common Subsequence
4. Edit Distance
5. 0 - 1 Knapsack problem
6. Min sum path in the matrix
7. Unique Paths using DP
8. Climbing Stairs
9. Min Jumps to reach end

10. Maximum-sum-such-that-no-two-elements-are-adjacent

11. Longest palindromic subsequence

Misc

1. Activity Selection Problem | Greedy Algo-1
2. Job Sequencing Problem
3. Connect n ropes with minimum cost
4. Minimize Cash Flow among a given set of friends who have borrowed money from each other
5. Count InversGoogle | Onsite | Min Transactions - LeetCode Discussions in an array | Set 1 (Using Merge Sort)