Linked List

> Basic Operations on Single Linked List

> Doubly Linked List

> Circular Linked list

> Reverse a Linked List - ( Recursion, Stack, Iterative)

> Find middle of a single Linked List

> Find nth Node from end

> Add first & last , second & second last ...... in O(n) - ( Recursive and Non recursive)

> Linked List is Palindrome or not

> Detect Loop

> Remove Loop

> Merging Point of Linked Lists ( slow and fast, hashing)

Stacks

> Stack implmentation using Array - Basic Operations

> Stack implementation using LinkedStack - Basic Operations

> Multiple stacks in 1 Array

> Min element in a Stack in O (1)

> Implement Stack using Queues

> Brackets Validation

> HTML TAGS validator

Queues

> Queue Basic Operations - (ArrayQueue , LinkedQueue, Circular Queues)

> Implement Queue using Stacks.

Hash Table

> Implement Hash Table with collision handling

> Count the number of words in a string

Priority Queues and Heaps

> Find median in a stream

> Operations on Binary Min heap

> Kth largest element in a stream

> Merge K sorted linked list

Greedy Algorithms

> Knapsack

> Job Sequencing

> Interval Scheduling

> Minimum number of Platforms required for a railway/bus station

Backtracking

> Generate all Binary strings of length n

> Print all permutations of a given string

> N-Queen Problem

> Rat in a Maze Problem

Sorting

> Merge Sort (recursive and non-recursive)

> Quick Sort (recursive and non-recursive)

> Heap Sort

String Algorithms

> KMP

> Z Algorithm

Binary Search

> Binary Search - Recursive and Non recursive

> Find all pairs with sum K

> Find first occurence of an integer in a sorted list with duplicates

> Rotation count of a sorted array

> Search element in a rotated sorted array

> Find missing element

Trees

> Creating a BST

> Traversals (Inorder, Preorder and Postorder).

> Iterative Traversals (Inorder, Preorder and Postorder).

> Calculate Height of a Tree

> Path in the Tree ( Highest cost , Print the paths)

> Level Order Traversal

> Level Order Traversal in Spiral Form

> Reverse Level Order Traversal

> Mirror of a Tree

> Vertical Order Traversal

> Top View of a Tree

> Bottom View of a Tree

> Check is tree is valid BST or no

> Binary Tree to Doubly Linked List

> Delete a Node in BST

> Palindromic tree

> Lowest Common Ancestor

> Diameter of a Binary Tree

> Delete a Tree

> Construct a Tree from Inorder and Preorder,

> Construct a Tree from InOrder and PostOrder

> Implement Tree using Array

Suffix Arrays

> Longest Repeated String - Overlapping

> Longest Repeated String - Non - Overlapping

> Longest Repeated String which occurs n times

> All Anagrams in Dictionary

> Longest Common Prefix

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Balanced Trees

> AVL ( Implementation Optional )

> Red-Black trees ( Will be discussed in tomorrow's class)

> Tries (Will be discussed in tomorrow's class)

Graphs

> Graph Implementation ( Vertex List using the Hash table )

> Edge List

> Adjacency Matrix

> Adjacency List

> Traversals - DFS & BFS

> Dijkstra's Shortest Path algorithm

> Floyd-Warshall algorithm

> Kruskal's algorithm

> Prim's algorithm

Dynamic Programming

> 0-1 Knapsack problem

> Longest Common Subsequence

> The subset sum problem

> Longest Increasing Subsequence

> The coin change problem

> Count number of ways to cover a distance

> Matrix Chain Multiplication

> Edit Distance problem for strings

> Cut the rod to maximize profit

> Minimum Jumps to reach the last element of array

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Matrix:

* Combination of all paths
* Spiral matrix