Data Structure Algorithms

* Sorting
* LinkedList
* List
* Spanning tree
* Tree
* Graph
* Stack
* Hashing
* Heaps
* Hash
* Maps

Day -1 and Day -2 ( Math & Programming Language)

* Set operators
* Groups
* Powersets
* Relations

Languages – Java / Python / C / C++

Day – 3 & day -4 [ Array Rotation]

* Array Rotation
* Find duplicates in array
* Missing numbers in a array
* Merge two sorted arrays
* Subarray with given sum

Day -5 & Day -6 [ Matrix]

* Rotate Matrix elements
* Matrix Multiplications
* Magic Square
* Square of Matrix diagonal elements

Day 6 to Day -9 [ STRINGS]

* Changing -parsing -slicing -pasting – analysing strings
* Word count in strings
* Count distinct subsequences
* Sparse search
* Kth non- repeating character
* Anagram substring search

Day 10 – 11 [SEARCH and SORTING]

* Searching algorithms
* Sorting algortihms
* 2 pointers

Day 12 and Day 13 [ BIT Magic and Recursion]

* Bit patterns or bit numerals
* Sum of bitwise XOR
* Least significant K bit
* Same weight Integer
* Recursion
* Tower of Honai
* Inorder / Preorder / Post Order Tree
* Traversals
* DFS of Graph
* Tail recursion

Day -14 and Day 15 [ Linked list]

* Used in implementing stack and queues and in graphs
* When data entries are huge and cant handle on their own ( Arrays)
* Middle element of LL
* Merge 2 LL
* Rotate doubly LL by n nodes
* Josephus circle
* Binary Tree to circular doubly LL

Day 16 – Day 18 – Stack

* Infix and postfix operation
* Queue using stacks
* Prefix to infix conversations
* Prefix to postfix conversions
* Postfix to prefix conversions
* Expression evaluations

Operation on Stack

* Reverse a stack using recursion
* Delete middle element of stack
* Balanced expression with replacement
* Bubble sort using 2 stacks

Day 19 – Queues

* Reversing the first K elements of queue
* Priority Queue using linked list
* Implement stack and queue using deque
* Reverse a path in BST using queue
* Zig-Zag Tree Traversal

Day 20 and Day -21

* AVL Trees , Red -Black Trees , self balanced tree
* In order Tree Traversal without recursion
* Inorder successor of node in binary tree.
* Reverse tree path.
* Morris traversal for Pre-order
* Check if a given binary tree is sum tree
* Check if 2 nodes are cousins in binary tree.

Day 22 [BST]

* Sorted Linked List to balanced tree
* Convert BST to min heap
* Merge 2 BST’s with limited extra space
* Lowest common ancestor in BST
* A program to check if binary tree is BST or not.
* Second largest element in BST
* Sum of K smallest elements in BST

Day -23 and Day 24 [HEAP]

* Heap is tree based data structures
* 2 types of HEAP 🡪 Minimum heap & Maximum Heap
* Prims Algorithms
* Dijkstra’s Algorithms
* Priority Queue efficiently implemented using heap data structure
* Leaf starting point in binary heap data structure
* Maximum difference between 2 subsets of m elements
* Maximum distinct elements after removing k elements
* K maximum sum combinations from 2 arrays
* Convert min heap to max heap
* Sort a nearly sorted ( or k sorted) array

Day – 25 and Day -26 [ GRAPH]

* Non -linear data structure
* Topological sorting
* Graph used to solve real-time problems
  + Shortest path between 2 cities
  + N-Queens Problems
  + M- coloring graph problems
* Longest path between any pair of vertices
* Longest path in directed acyclic graph
* Kahn’s Algorithms for topological sorting

Basic Graph Problems

* Transpose graph
* Count numbers of trees in a forest
* Print all paths from a given source to destination using BFS
* Minimum edge reversal to make a root.
* A matrix probability questions

Complex Graph problems

* Detect cycle is directed graph
* Tug of war
* N- Queens problem
* M coloring problem
* Hamiltonian Problem
* Bellman -ford algorithm
* Floyd Warshall Algorithms
* Dials Alogorithms
* Dijkstra’s shortest path Algorithms
* Rat in a Maze
* The knight – tour problem

Day 27 and Day -28 ( Greedy algorithms )

* Used in operating systems
* Huffman coding
* Job sequence problem
* Water connection problem
* Policeman catch thieves
* Minimum swaps for bracket balancing
* Fitting shelves problem

Complex greedy algorithms

* Kruskals minimum spanning tree
* Prim’s Minimum spanning tree
* Graph coloring
* K-centers problem
* Shortest supersting problem

Day -29 and Day -30

* Dynamic programming

Basic DP problems

* Fibonacci numbers
* Nth Catalan numbers
* Compute ncr % p
* Print Fibonacci serries in reverse order
* Longest common subsequence
* Longest repeated subsequence
* Minimum sum path in triangle

Intermediate DP problems

* Floyd warshall algorithms
* Bellman-ford algorithms
* 0-1 knapsack Algorithms
* Egg dropping puzzle
* Dice Throws problem
* Word break Problem
* Vertex cover Problem
* Tile stacking problem
* Box -stacking algorithms

Complex DP problems

* Palindrome partitioning
* Word wrap problem
* Matrix chain manipulation
* Maximum sum bitonic subarray
* Maximum sum rectangle in 2D matrix
* Number of subsequence in a string divisible by n;
* Printing shortest common super sequence.