

DSA

Introduction

Time Complexity
Space Complexity
Types Of Data Structures

Declaration of array

Add and updating 1d array
Print Array
Remove Elements from Array
Reverse an array
Find Minimum of array
Find Max of array
Find 2nd max of array
How to resize an array
Check Palindrome

Introduction of Singly LinkedList

Implementation of Singly linkedlist
Print Head
Insert Newnode
Print Nodes
Find Length of Linkedlist
Insert at Beginning
Insert at End
Insert at given position
Delete first Node
Delete last Node
Delete at given position of Linkedlist
Search For element
Reverse a linkedlist
Find middle node
Find nth node from end
Remove duplicates in the sorted linkedlist
Remove the given key
Merge two sorted Linkedlist

Represent Doubly linkedlist

Print Elements of Doubly LinkedList
Insert at beginning of doublylinkedlist
Insert at end of DLL
Delete first Node of DLL
Delete Last Node of DLL

Implement Circular Linkedlist

Print a Circular Linked List
Insert at Start of CLL
Insert at End of CLL
Remove first Node of CLL

Implement Stack

Stack using Linkedlist

Stack using Array

Reverse a string using stack

Implement Queue

Enqueue

Dequeue

Implement Binary Tree

Recursive Preorder

Iterative Preorder

Recursive Inorder

Iterative Inorder

Recursive Postorder

iterative Postorder

Level Order

Find max value of Binary Tree

Represent binary tree in java

Insert value to Binary Tree

Search a given Key

Introduction of priority Queue and Binary Heap

Represent Binary Heap

Implement Max binary Heap

Bottom up Reheapify (SWIM) in Maxheap

Insert in maxheap

Top Down ReHeapify(SINK)

Delete max element in MaxHeap

Linear Search

Binary Search

Search Insert Position in sorted Array

Bubble Sort

Insertion Sort

Selection Sort

Merge two sorted Arrays

Merge Sort

Quick Sort

Squares of a Sorted Array

Introduction of Graphs

Adjacency Matrix

Implementation of adjacency Matrix of undirected Graph

Adjacency list of Undirected Graph

Breadth First Search (BFS)

Iterative Depth search First (DFS)

Recursive DFS

Connected Components in undirected Graph

Introduction to Hashing

What is Hash Function

Intro to Hash Table

Separate Chaining Collision Resolution

Represent hashNode in Hash Table

Implement Hash Table

Implement Hash Table Separate Chaining

Put Key value pair in hash Table

Get value by key

Remove Key

Introduction to Intervals and Overlapping Intervals

Merge Intervals

Intro to Trie DS

Represent Trie Node

Implement TrieNode

Insert a Word in Trie