Linked list: is a linear data structure

Types:

1.singlt linked list:can be traversed only in one direction.

Linear data structures (elements occur in sequence)

1.array

2.linked list :

head/tail

3.stack

4.queue

**Trees** are non -linear data structures data is not stored in sequence.

In trees top node is called root node:

Each node contains some data.

Each node contains some data and linked or referenced to the next node …called children

Tree is also known as recursive data structure.

**Number of edges:**

1.If N nodes

Then there would be n-1 edges

Binary tree has logN time for searching element.

Heap is a tree data structure. Used to implement priority queues.

In general tree nodes are un ordered.

Binary tress means node has at least two children

**Binary search trees:**

One node is connected to “n” number of nodes.

BINARY SERACH TREE is a node-based data structure.

1.Every node in the left sub-tree must have the value less than root node.

2.every node in the right sub -tree must have the value greater than root node.

🡪Pre-order Traversal—1st time visit

Node

Left

Right

🡪In-order Traversal-2nd time visit

Left(empty)

Node

Right(right empty)

First we will check till left==null;

Next we will check till right ==null;

🡪Post-order Traversal-3rd time visit.

Left

Right

Node

1.use stack data structure

2.check current element in stack and pop it and add it to the result variable

Recursion:

It is a technique where a function calls itself.

In-order vs pre order:

Pre-order: root, left, right

In-order :left, root, right

If we can find root In in-order traversal, left part of root is related to left sub -tree

Right , part of the root is related to right-sub tree

Ascending order: small to big

1st position: where we are in the tree (1st smallest/2nd smallest). Position.

2nd position: the value we want to return.

1. [Validate Binary Search Tree](https://leetcode.com/problems/validate-binary-search-tree/)

Root should be smaller than right- sub tree

and larger than left sub tree

methods:

1.recurssion

2.take min ,max and compare them with left and right sub trees.

[102. Binary Tree Level Order Traversal](https://leetcode.com/problems/binary-tree-level-order-traversal/)

Method used:

1.take result array

2.Take a queue

3.check for current node

5.pop every node and push to the current node,

6.add current level\_node to result;

Hashmap:

Stores the values in pairs as key-value

Put method is used to add element in the Hashmap

Hashset:

Stores only object

Add method is used to add element in the Hashset

Sliding window technique:

LRU Cache:

1. 0(1) time complexity   
   queue  
   hasmap  
   doubly linked list

Head.next=n;

Head

|

^

4->8->12->null

Current

Node Current=head;  
current.next=head;

4->8->12  
h

Head.next=head.next.next;  
start of the doubly linked list

New node.next=head;  
head.prev=new node(5);

newnode.prev=null;

head=new node;

Matrix  
transpose matrix:columns into rows

Heap:

Min heap: root node is the smallest   
max heap: root node is the biggest  
  
Heaps are used to implement priority Queues.

Min Heap Is used to determine the shortest path between vertices in a graph

Heap is a data structure is a complete binary tree.//heap is also called binary heap

Heap is represented in as an array.

Parent node: A[(i-1)/2]

Left child node: A[(2\*i)+1]

Right Child Node: A[(2\*i)+2]

In -order

Pre -order

Level -order tarversal