Data Structures Revision

Searching Techniques

Searching Techniques in Arrays

Following are the classes in which searching techniques lie:

1. Sequential Search
2. Interval Search
3. Hashing

Searching Techniques: Categorization

We have studied the following searching techniques

1. Sequential Search
   a. Linear Search
2. Interval Search
   a. Binary Search
3. Hashing

Linear Search

Searching Techniques: Linear Search

Time Complexities of Linear Search is:

	Best	Average	Worst
Linear Search	O(1)	O(n/2)	O(n)

Searching Techniques: Linear Search

Space Complexity of Linear Search is:

	Worst
Linear Search	O(1)

Binary Search

Searching Techniques: Binary Search

Binary Search follows the divide and Conquer Strategy.
Data should be sorted before applying Binary Search.
Time Complexity of Binary Search is:

	Best	Average	Worst
Binary Search	$O(1)$	$O(\log_2(n))$	$O(\log_2(n))$

Searching Techniques: Binary Search

Space Complexity of Binary Search is:

	Worst
Binary Search	O(1)

Searching through Hashing

Open VS Closed Hashing

Closed Hashing (Open Addressing)	Open Hashing
Linear Probing	Chaining
Quadratic Probing	
Double Hashing	
Hopscotch hashing	
Robin Hood hashing	
Last-come-first-served hashing	
Cuckoo hashing	

Searching Techniques: Hashing

Time Complexities of searching through hashing

Hashing	Best	Average	Worst
Insert	O(1)	O(1)	O(n)
Retrieve	O(1)	O(1)	O(n)
Delete	O(1)	O(1)	O(n)

Searching Techniques: Hashing

Space Complexity of hashing with Open Addressing is:

	Worst
Hashing with Closed Hashing	O(m)

Where m is the size of the table.

Searching Techniques: Hashing

Space Complexity of hashing with Chaining is:

	Worst
Hashing with Open Hashing	O(m + n)

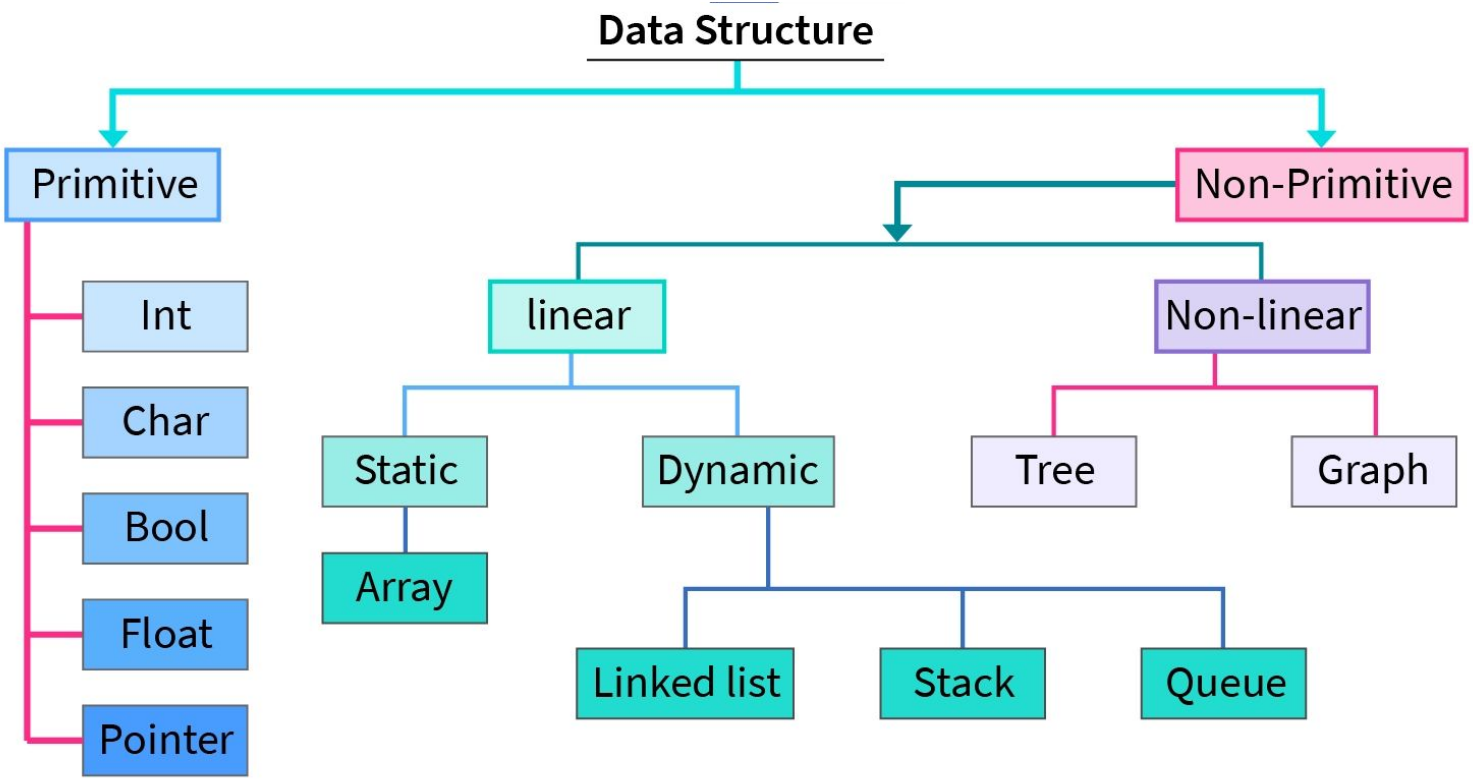
Where m is the size of the table and n is the size of input elements.

Hashing: Real Life Applications

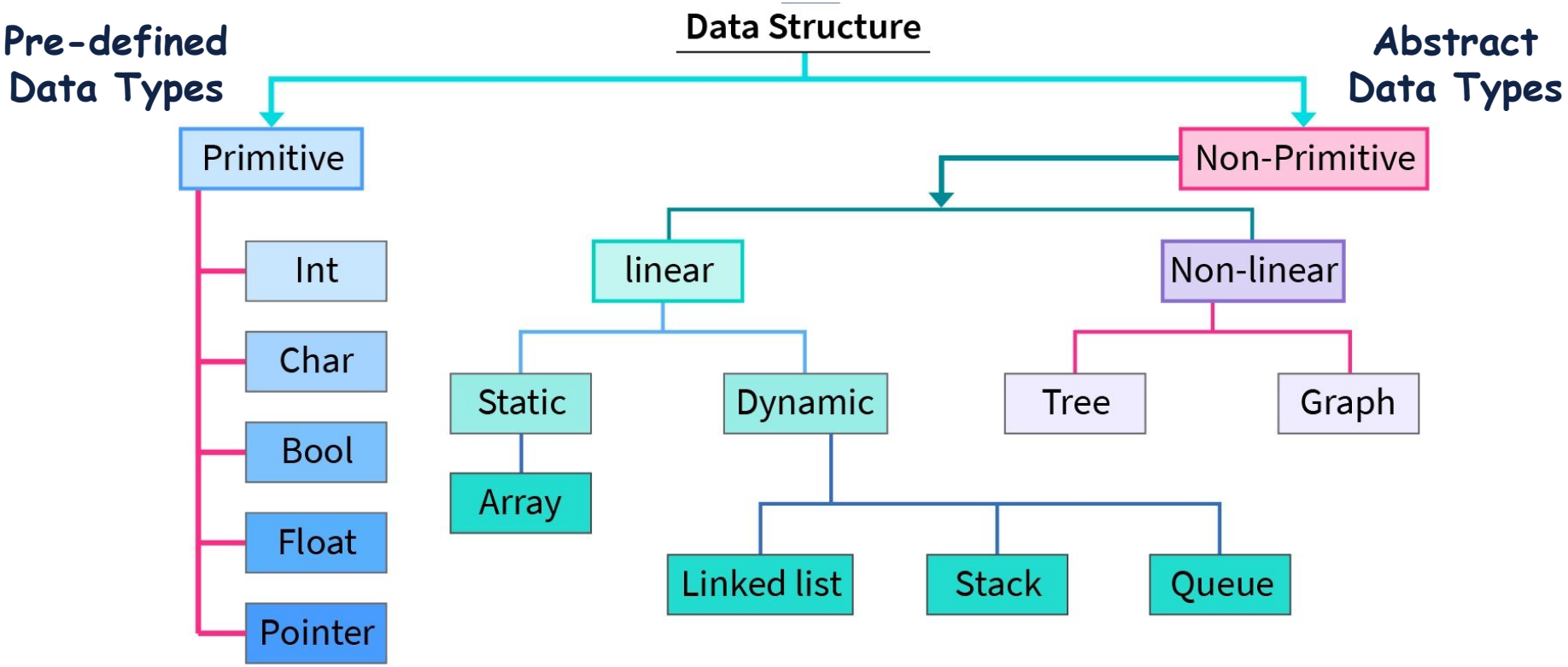
- Compilers use Hash Tables to keep track of declared Variables
- Hash Tables are used in Online Spelling Checkers. For detection of misspelling an entire dictionary is hashed and words are checked in constant time
- Games use hash table to store positions, thus saving computation time if same position is encountered again.
- Hash Tables are used to solves problems like check inequalities or find duplicates.

Data Structures

Data Structures



Data Structures



Arrays

Array

Time Complexities of Array are:

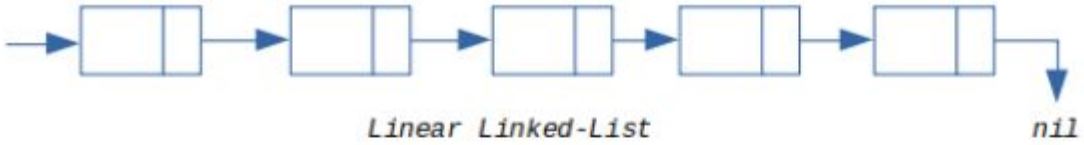
Array	Worst
Insert	O(n)
Search	Depends on the Searching Algorithm
Delete	O(n)

Array: Uses of Array

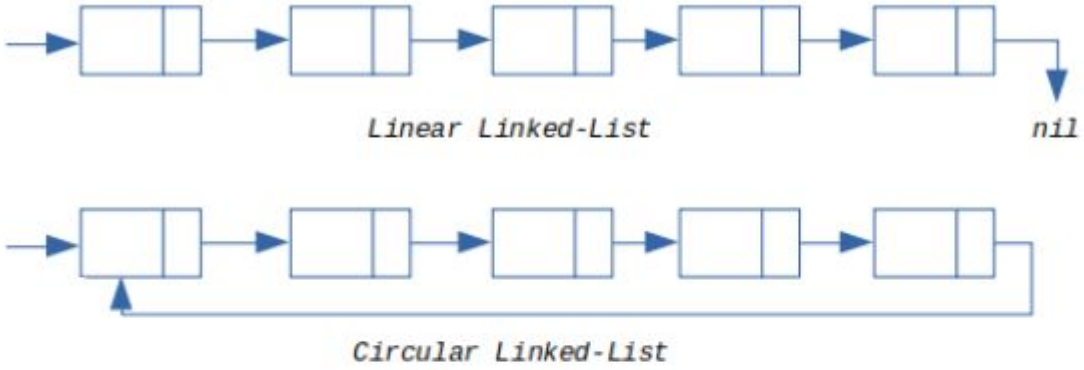
- Arrays are used as the base of all sorting algorithms.
- Arrays are used to implement other DS like a stack, queue, etc.
- Used for implementing matrices.
- Data structures like trees also sometimes use the array implementation since arrays are easier to handle than pointers.

Linked Lists

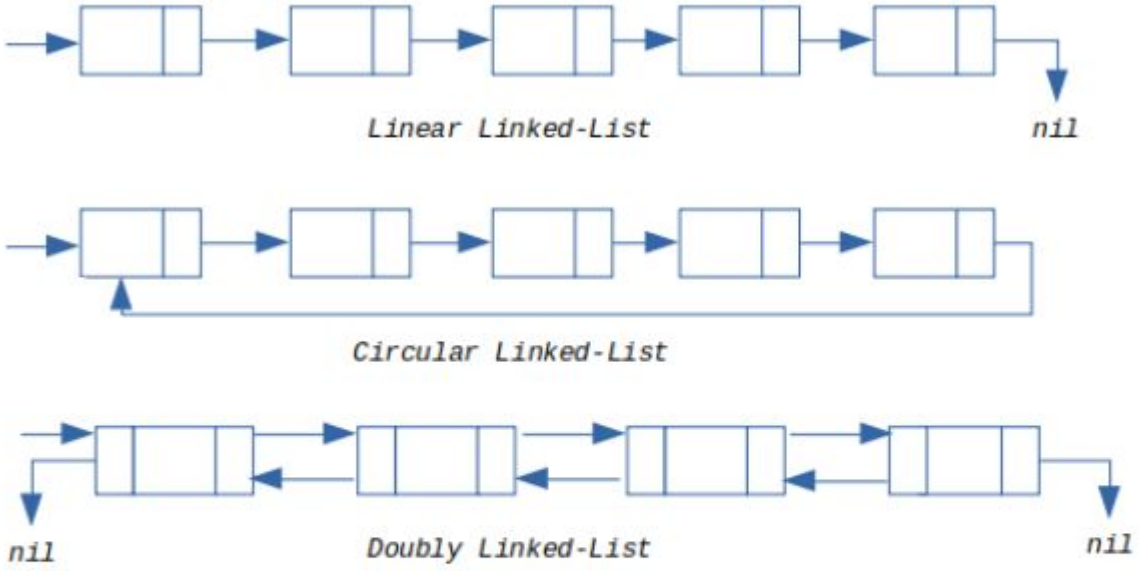
Linked Lists: Singly Linked List



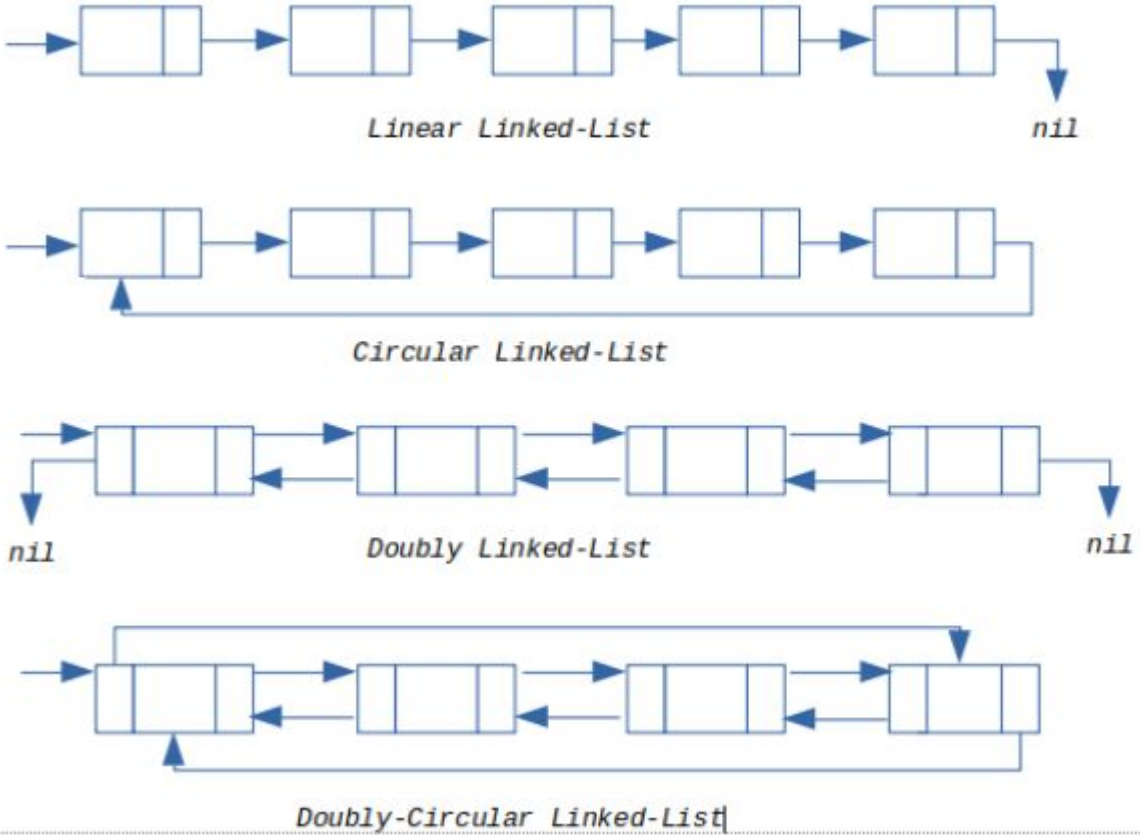
Linked Lists: Circular Linked List



Linked Lists: Doubly Linked List



Linked Lists: Doubly-Circular Linked List



Linked Lists

Time Complexities of Singly Linked List are:

Singly Linked List	Worst
Insert at Start	O(1)
Insert at End	O(n)
Delete at Start	O(1)
Delete at End	O(n)
Search	O(n)

Linked Lists: Points to Ponder

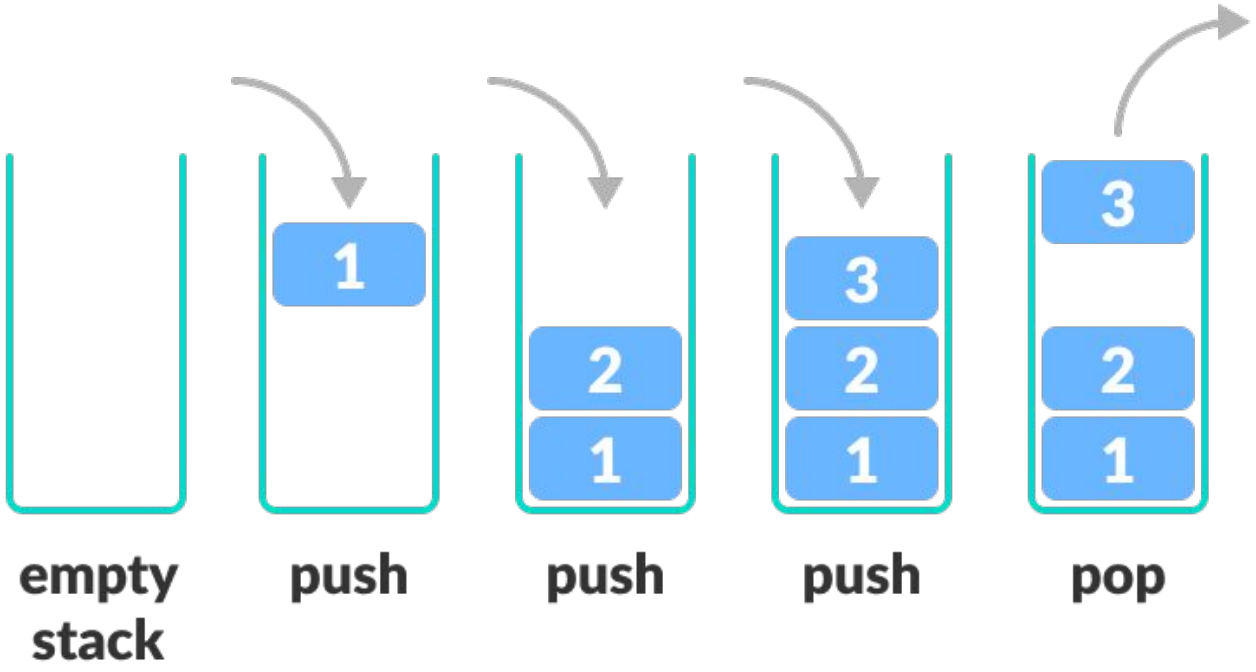
1. How to Find the middle of a given linked list?
2. How to apply Binary Search on Singly Linked List?
3. Why is removing the given node from a doubly-linked list faster than removing the given node from a singly-linked list?

Linked Lists: Uses of Linked Lists

- Implementation of stacks and queues
- Implementation of Tree and graphs
- Dynamic memory allocation: We use a linked list of free blocks.
- Performing arithmetic operations on long integers
- Manipulation of polynomials by storing constants in the node of the linked list

Stack

Stack: Last In First Out (LIFO)



Stack

Time Complexities of Stack are:

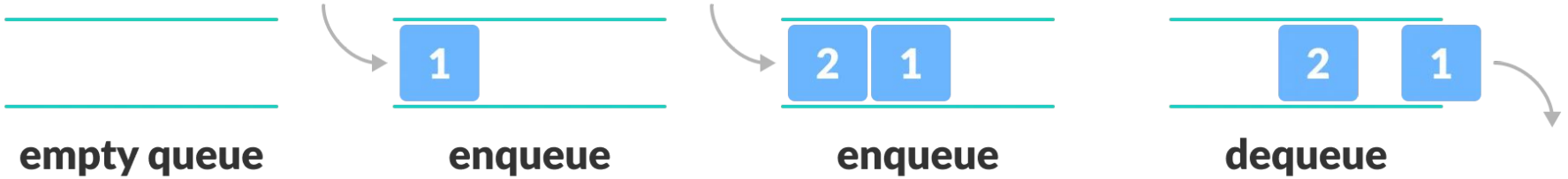
Stack	Worst
Push (Insert)	O(1)
Pop (Delete)	O(1)

Stack: Uses of Stacks

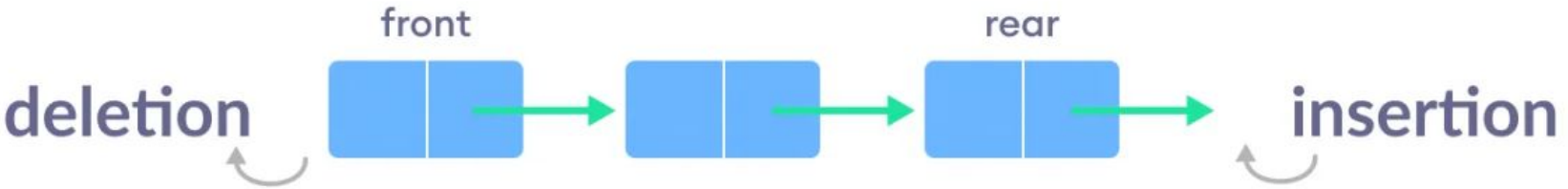
- Converting infix to postfix expressions.
- Undo/Redo button/operation in word processors.
- Syntaxes in languages are parsed using stacks.
- Forward-backward surfing in the browser.
- History of visited websites.
- Loading bullets into the magazine of a gun. The last one to go in is fired first.
- Recursion.
- Used in IDEs to check for proper parentheses matching

Queue

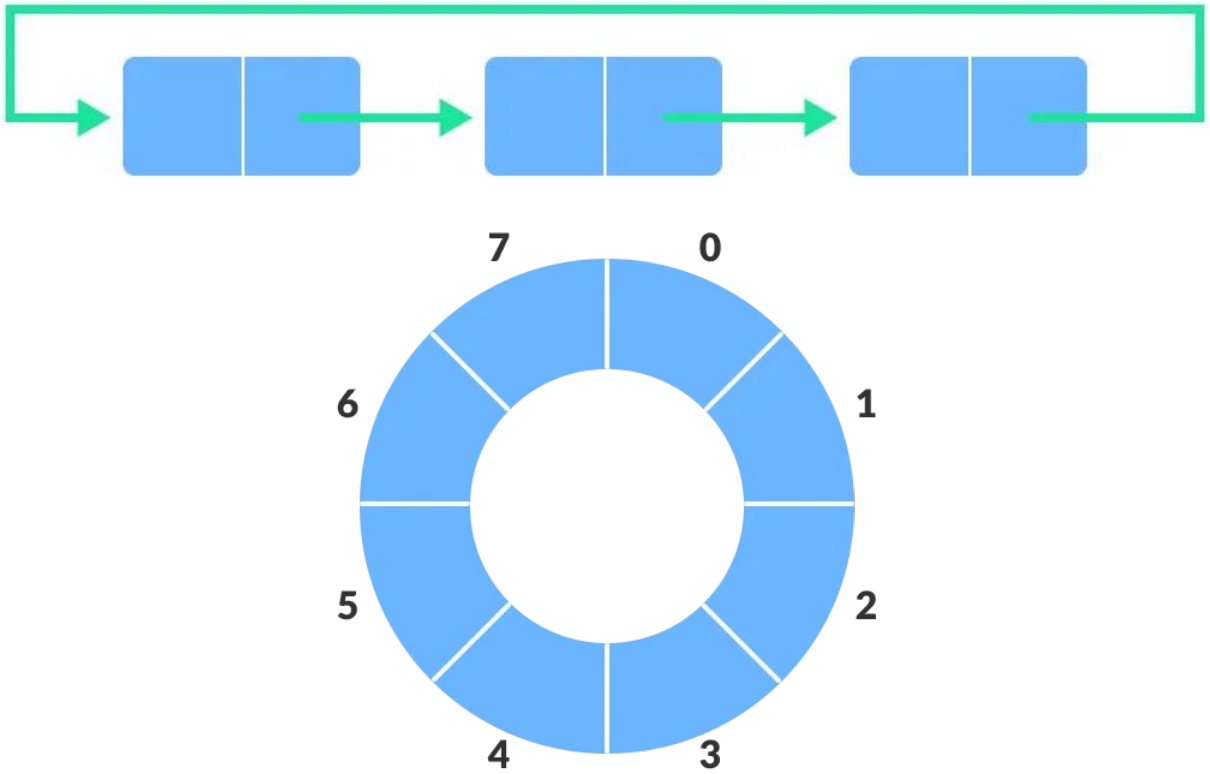
Queue: First In First Out (FIFO)



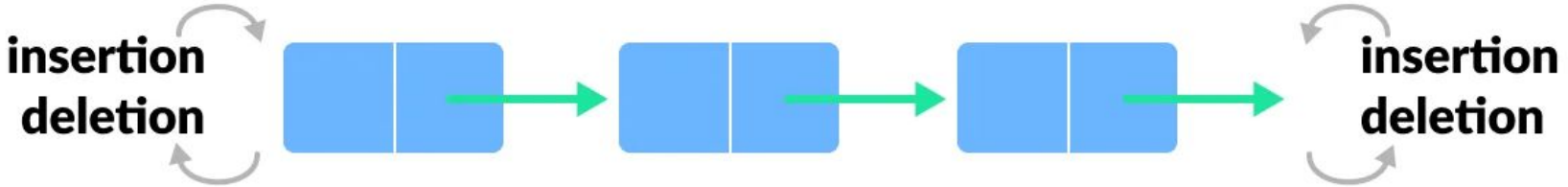
Queue: Simple (Linear) Queue



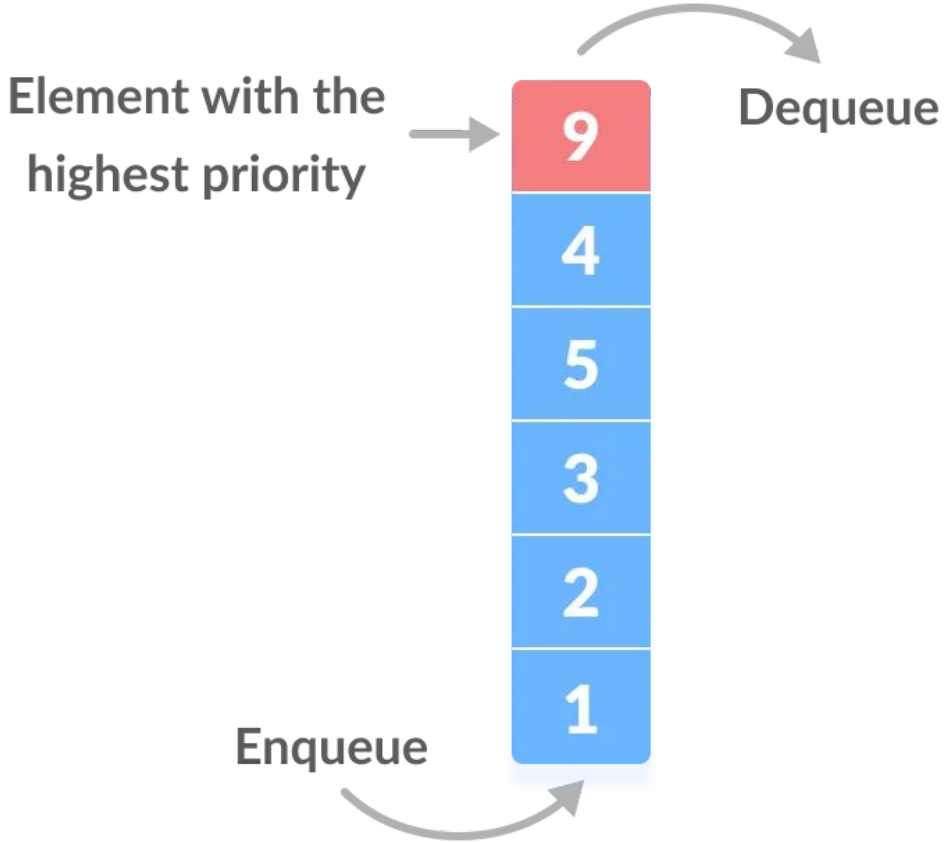
Queue: Circular Queue



Queue: Double Ended Queue (DeQueue)



Queue: Priority Queue



Queue

Time Complexities of Queue are:

Queue	Worst
Enqueue (Insert)	O(1)
Dequeue (Delete)	O(1)

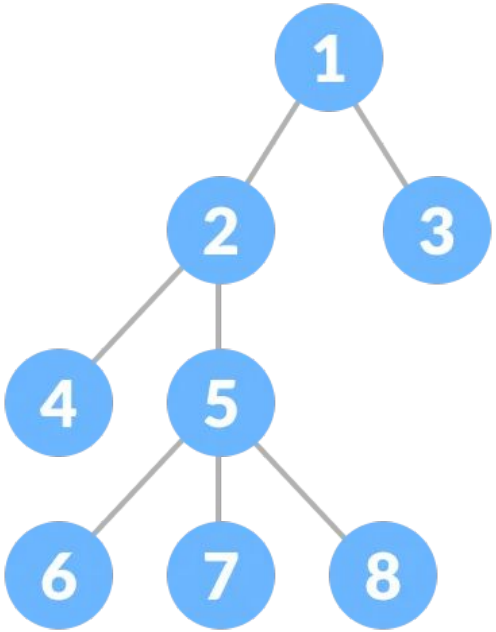
Queue: Uses of Queue

- Operating System uses queues for job scheduling.
- Sending an email, it will be queued.
- While switching multiple applications, windows use circular queue.
- A circular queue is used to maintain the playing sequence of multiple players in a game.
- A queue can be implemented in LinkedList-based Queue, Array-based Queue, Stack-based Queue.
- Priority queues are used in file downloading operations in a browser

Trees

Trees: Non-Linear Data Structure

A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges.



Traversals: Binary Trees

There are 2 types of traversals

1. Breadth First Traversal (Queue)
    a. Level Order Traversal

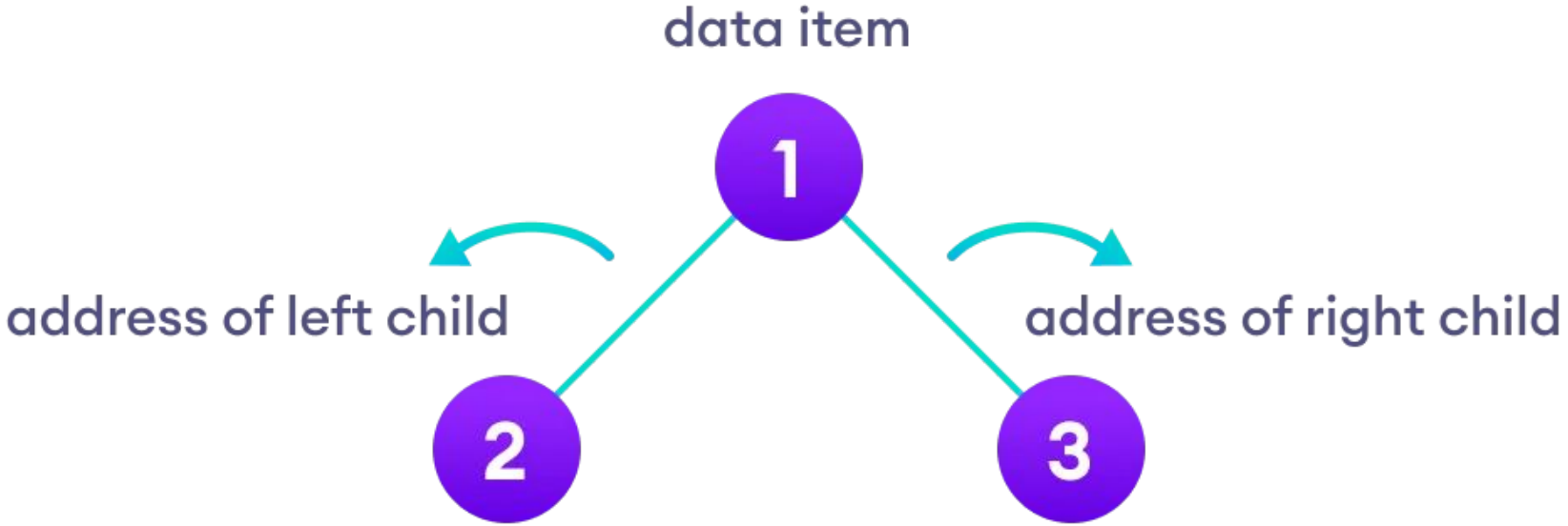
2. Depth First Traversal (Stack)
    a. Pre-Order Traversal
    b. In-Order Traversal
    c. Post-Order Traversal

Trees: Uses of Trees

- To store the possible moves in a chess game.
- To store the genealogy information of biological species.
- Store the Hierarchical information of the Drives and Folders in Computer.
- The decision-based algorithm is used in machine learning which works upon the algorithm of the tree.

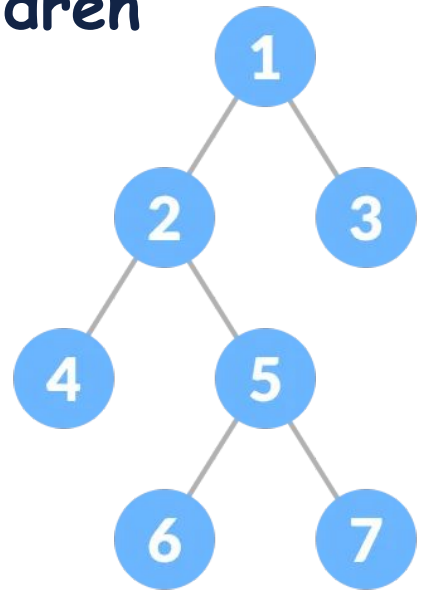
Trees: Binary Trees

A binary tree is a tree data structure in which each parent node can have at most two children.



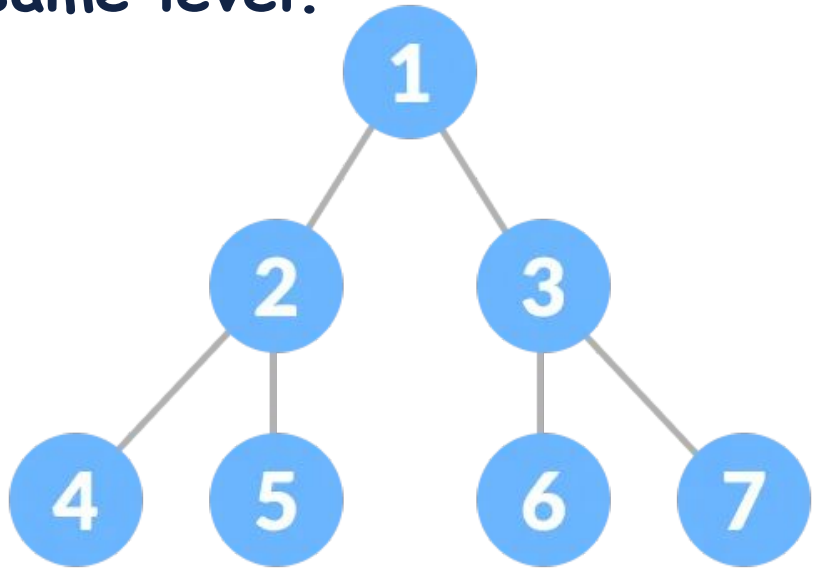
Types of Binary Trees: Full Binary Tree

A full Binary tree (aka proper binary tree) is a type of binary tree in which every parent node/internal node has either two or no children



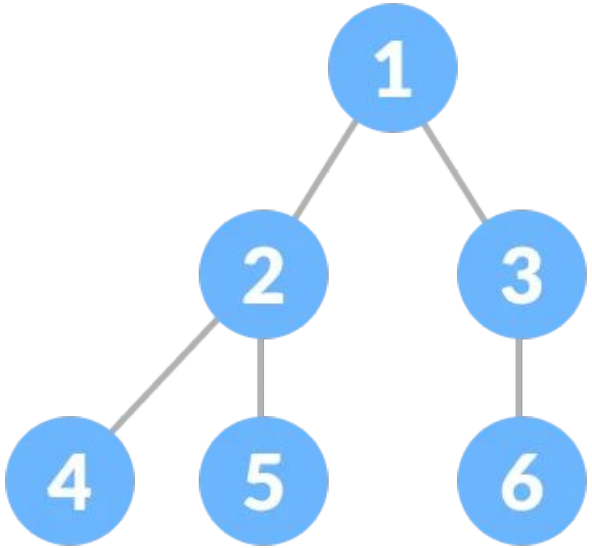
Types of Binary Trees: Perfect Binary Tree

A perfect binary tree is binary tree in which every internal node has exactly two children and all the leaf nodes are at same level.



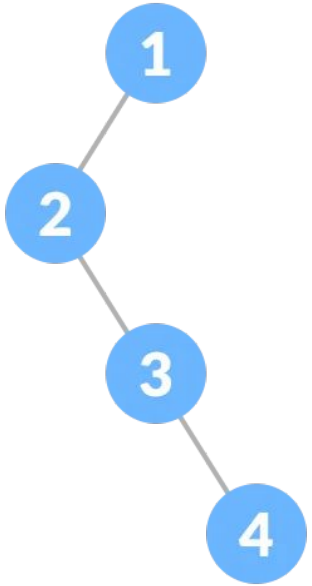
Types of Binary Trees: Complete Binary Tree

A complete binary tree is a binary tree in which all the levels are completely filled except possibly the lowest one, which is filled from the left.



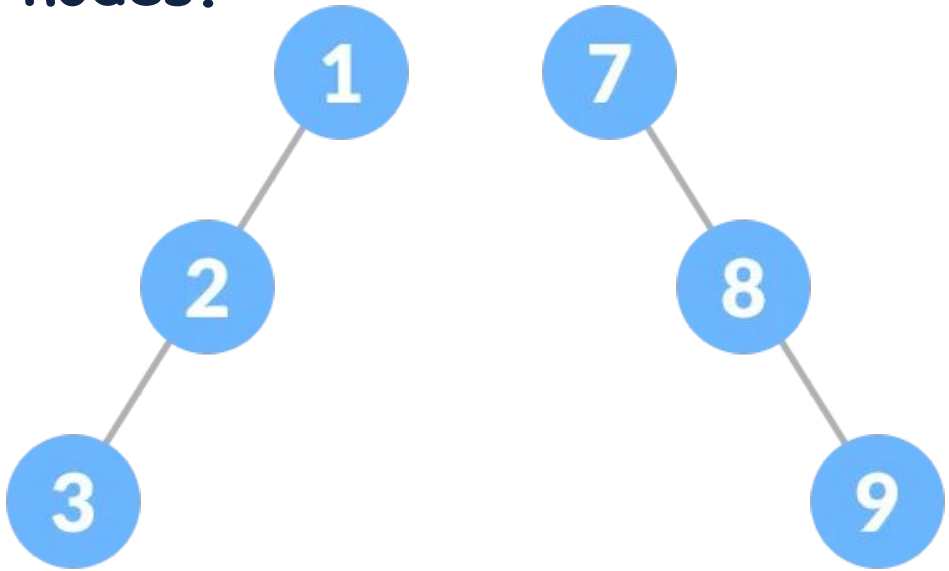
Types of Binary Trees: Degenerate Binary Tree

A degenerate or pathological tree is the tree having a single child either left or right.



Types of Binary Trees: Skewed Binary Tree

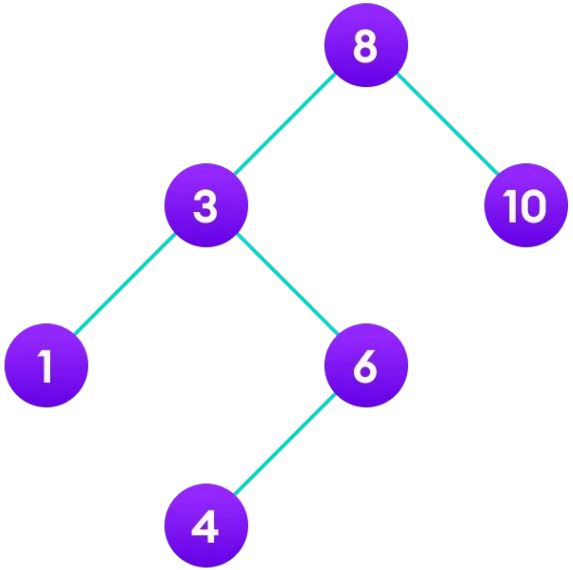
A skewed binary tree is a pathological/degenerate tree in which the tree is either dominated by the left nodes or the right nodes.



Binary Search Trees

Types of Binary Trees: Binary Search Tree

All nodes of left subtree are less than the root node and all nodes of right subtree are greater than the root node.



Binary Search Trees

Time Complexities of Binary Search Trees are:

Binary Search Trees	Worst
Insert	O(n)
Traverse	O(n)
Delete	O(n)

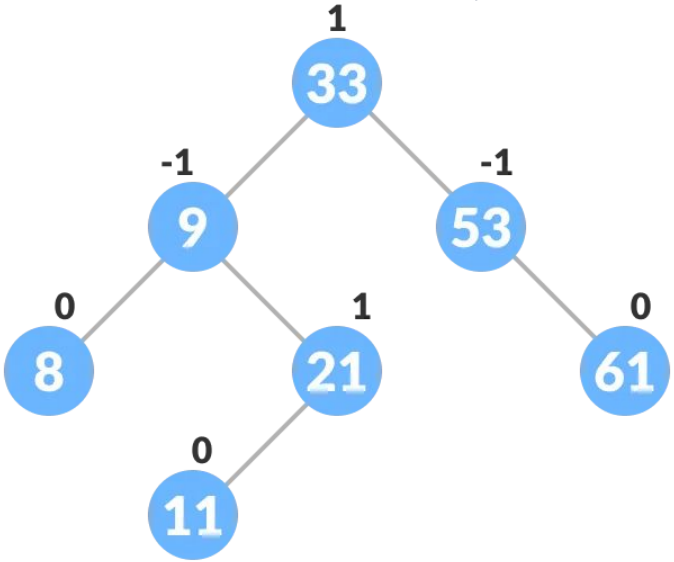
Binary Search Trees: Uses of BST

- They are also helpful to implement various searching algorithms.
- It is helpful in maintaining a sorted stream of data.
- Used in many search applications where data is constantly entering/leaving, such as the map (ordered) and set objects in many languages' libraries.
- Used in almost every 3D video game to determine what objects need to be rendered.

AVL Trees

Types of Binary Trees: AVL Tree

AVL tree is a self-balancing binary search tree in which each node maintains extra information called a balance factor whose value is either -1, 0 or +1.



AVL Trees

Time Complexities of AVL Trees are:

Binary Search Trees	Worst
Insert	$O(\log_2(n))$
Traverse	$O(n)$
Delete	$O(\log_2(n))$

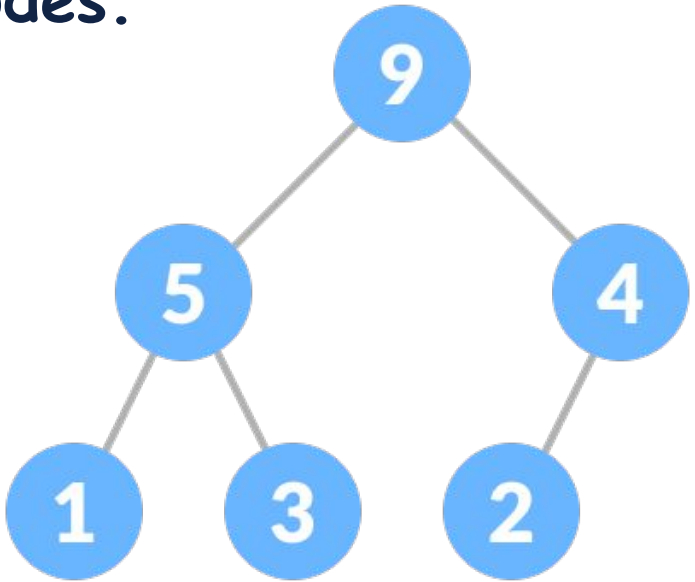
AVL Trees: Uses of AVL

- More Search and less Insertion/Deletion.
- Data Analysis and Data Mining and the applications which involve more searches.

Heap

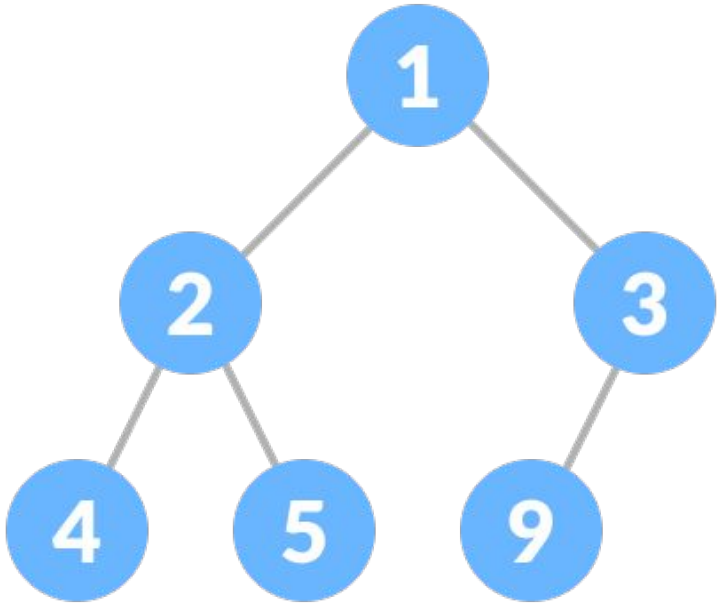
Types of Binary Trees: Max Heap

Max Heap is a complete binary tree that satisfies the heap property, where any given node is always greater than its child nodes.



Types of Binary Trees: Min Heap

Min Heap is a complete binary tree that satisfies the heap property, where any given node is always less than its child nodes.



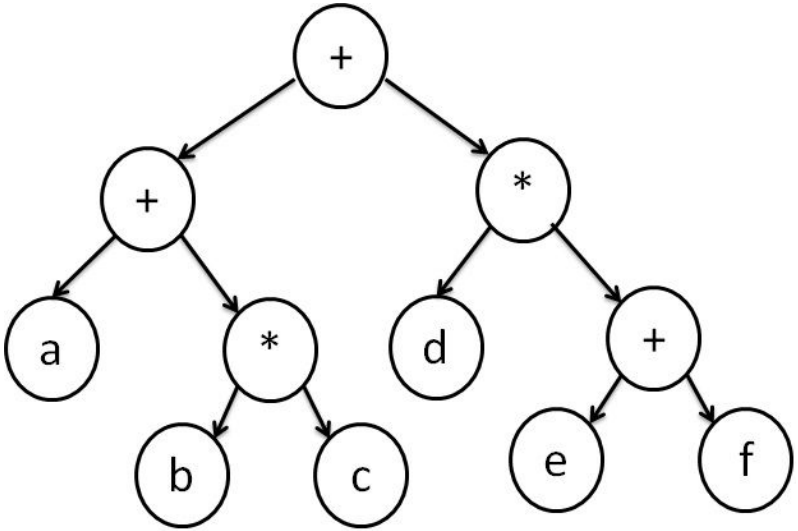
Heap: Uses of Heap

- Heaps are used to implementing a priority queue where priority is based on the order of heap created.
- If we are stuck in finding the Kthsmallest (or largest) value of a number then heaps can solve the problem in an easy and fast manner.

Expression Trees

Expression Trees

The expression tree is a binary tree in which each internal node corresponds to the operator and each leaf node corresponds to the operand



Learning Objective

Students should be able to perform Extraordinary Well in the Data Structures and Algorithm Exams.