syntax of expressions program block etc before translating into low level code.
- It is also used in memory management.

Infix Notation:

We write expression in infix notation, e.g a-b+c, where operators are used in between operands. It is easy for us to understand: but in an algorithm to process infix notation could be difficult and costly in terms of time and space consumption.

Prefix (Polish) Notation:

It is a way to write an expression without parenthesis in which operators are placed before it operands.

Scanned with CamScanner

Date

→ Advantage:

• The fundamental property of polish notation is the order in which the operations are to be performed is completely determined by the position of the operators and the operands.

Postfix (Reverse-Polish) Notation:

In post fix notation the operator symbol is placed after the operands it is also called reverse polish notation.

→ Advantages:

• Scanning is easy because operators are associated with operands.
• No need for parenthesis.
• Stalk implementation became easy.

Priorities Of Operators:

• Arthimatic operators
• Assignment operators
• logical operators

Stack Questions

Scanned with CamScanner

Date

6 2 3 + - 3 8 2 / + * 2 ↑ 3 +

Symbol	Op1	Op2	Value	Stack
6				6
2				62
3				623
+	2	3	5	65
-	6	5	1	1
3				13
8				138
2				1382
/	8	2	4	134
+	3	4	7	17
*	1	7	7	7
2				72
↑	7	2	49	49
3				3
+	49	3	52	52

E = (A * (B + C) * D)

Symbol	Stack	Expression
(	(	
A	(	A
*	(*	A
(	(*(	A
B	(*(	AB

Scanned with CamScanner

Date

+	(*(+	AB
C	(*(+	ABC
)	(*	ABC+
*	(*	ABC+*
D	(*	ABC+*D
)		ABC+*D*

- If symbol operator is equal to stack operator than push stack operator into output and push symbol operator into stack operator.

Symbol op = = stack op
↳ stack → o/p
↳ Symbol → stack

- If symbol operator is greater than stack operator push symbol operator into stack operator.

Symbol op > stack op
↳ push symbol op into stack

- If symbol operator less then stack operator than push stack operator into output and push symbol operator into stack.

Symbol op < stack op
push stack op → output
push symbol op → stack

Scanned with CamScanner

Date _____

(A*B /(A-C)+D*B)

Symbol	stack	Output	Comments
(	(		
A	(	A	(+* ✓
*	(*	A	(*+ X
B	(*	AB	priority ÷* same
/	(/	AB*	symbol ko stack m
(	(/(	AB*	r stack ko o/p m
A	(/(	AB*A	
-	(/(-	AB*A	
C	(/(-	AB*AC	
)	(/	AB*AC-	
+	(+	AB*AC/	shifted ↔
D	(+	AB*AC-/D	
*	(+*	AB*AC-/D	
B	(+*	AB*AC-/DB	
)		AB*AC-DB*+	AB*AC-DB*+

Recursive Function:

A procedure is said to be recursive if it calls itself and move towards the base criteria.

Recursive procedure may be defined by the two characteristics.

Scanned with CamScanner

Date

- Recursive procedure calls it self and at the base criteria where the procedure does not call itself.

- factorial is a simple resursive function.

5X4!) call
4! = 4X3!
0! → stop Base criteria

find f(8), f(2) = 9 If f(x) = 4f(x-3) - 1

f(8) = 4f(8-3) - 1
f(8) = 4f(5) - 1 → 1
f(5) = 4f(5-3) - 1
f(5) = 4f(2) - 1

putting the value of f(2) = 9

f(5) = 4(9) - 1
f(5) = 36 - 1
f(5) = 35

Using eq 1
f(8) = 4(35) - 1
f(8) = 140 - 1
f(8) = 139

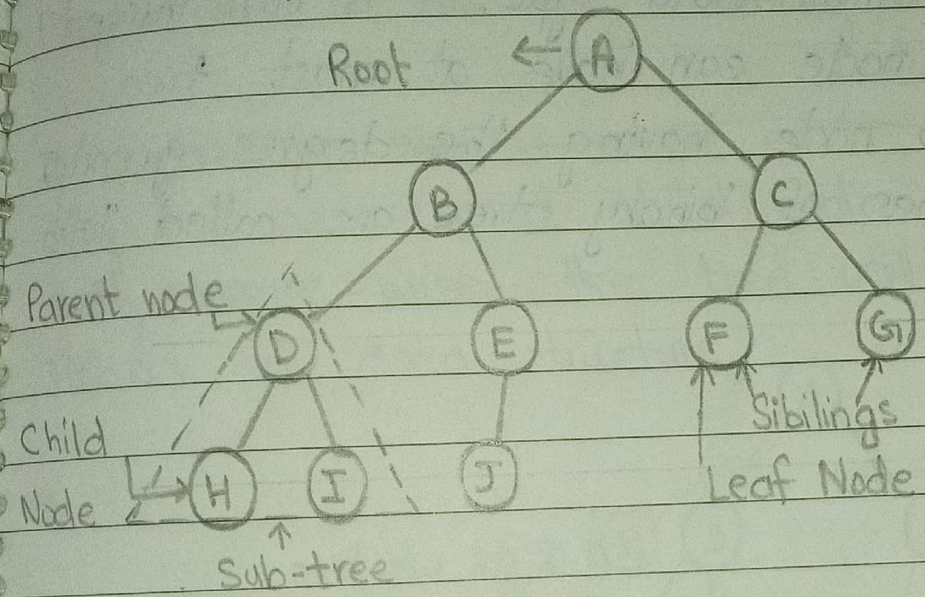
Q(a,b) = Q[(a-b,b)+1]
If a<b Q(a,b) = 0
If b ≤ a Q(a,b) = Q[(a-b,b)+1]

Scanned with CamScanner

Non-Linear Data Structure

Tree:

Tree can be defined as a collection of entities (nodes) linked together to simulate a hierarchy.



Basic Terminologies Used In Trees:

Parent: The immidiate predecessor.

Root: The node at the top of the tree is called root.

Path: Sequence of consecutive edges from source node to destination.

Depth of node: Length of path from root to that node.

Height of node: No of edges in the longest path from that node to leaf node.

Siblings: All the children of same parent are known as siblings.

Terminal Nodes: (Leaf / external / 0 degree nodes) The nodes having the zero degree (no child) is called terminal nodes.

Scanned with CamScanner

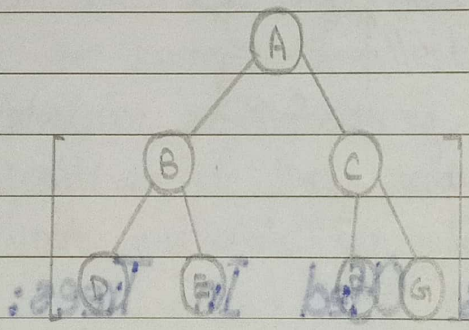
Date

Non-terminal nodes: The nodes having at least one degree is called non-terminal nodes. (non-leaf / internal / not zero degree nodes)

Child: The immidiate successor.

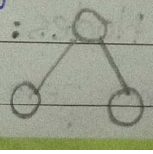
Binary Tree:

Binary tree is a abstract data type. It is constructed in a way that any node can have at most two branches (nodes). that is no node having the degree greater then two. The branches of binary tree are called left and right subtrees.



Technical Defination Of Binary Tree:

- A binary tree is a finite set of nodes either it is empty or consist of a root and two disjoints binary tree called left and right subtree.
- If every node can have two children or no children is called strictly binary tree. It is also called extended binary tree. Strictly binary tree with n-leafs contains (2n-1) nodes.



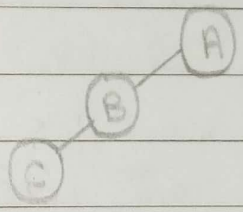
Scanned with CamScanner

Date _____

- The maximum number of nodes in a binary tree of depth K is $(2^K - 1)$ where the K must b greater or equal to 1 $(K \geq 1)$
- The maximum number of nodes in any level of binary tree is calculated as $(2^L - 1)$ L no of levels

Degenerate or skewed binary tree or diagonal tree:

A degenerate tree is where each parent node has only one associated child node. This means that performance-wise, the tree will behave like a linked list data structure.



Traversing is a process to visit all the nodes of a tree at least once.

Tree Traversing:

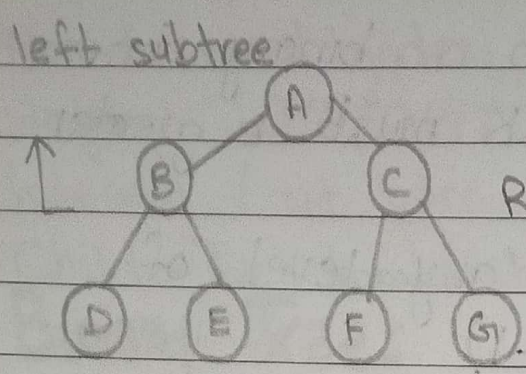
There are three ways which we use to traverse a tree.

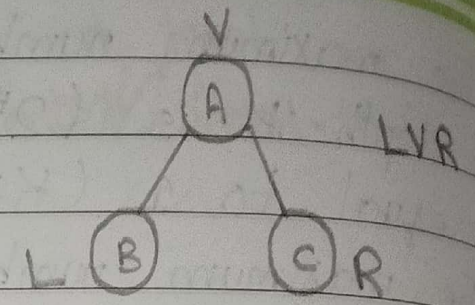
In-order Traversal:

In this traversal method, the left subtree is visited first, then the root and later the right sub-tree. We should always remember that every node may represent a subtree itself.

Scanned with CamScanner

Date

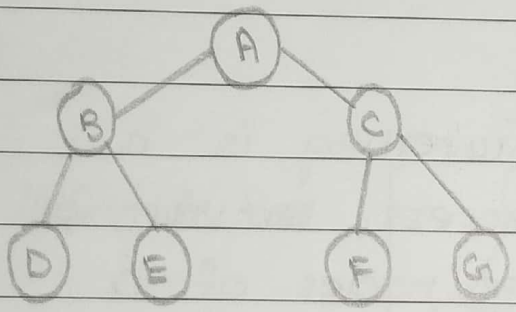


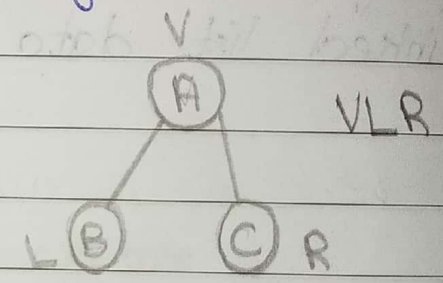


D → B → E → A → F → C → G

Pre-Order Traversal:

In this traversal method, the root is visited first, then the left subtree and finally the right subtree.





left before right

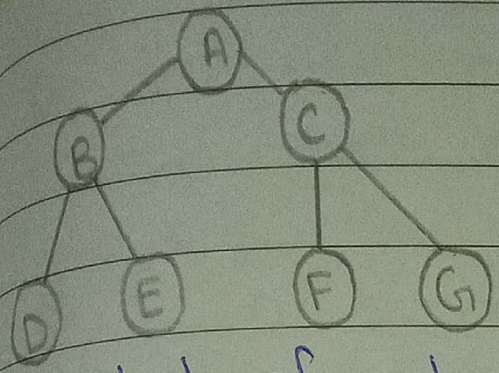
The output of pre-order traversal of this tree will be.

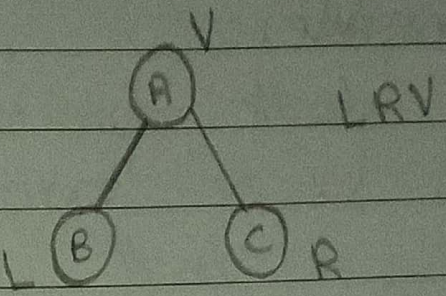
A → B → D → E → C → F → G

Post-order Traversal:

In this traversal method, the root node is visited last. First we traverse the left subtree, then the right subtree and finally the root node.

Scanned with CamScanner



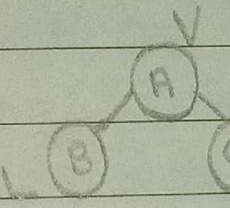


The output of post-order traversal of this tree will be.

$$D \rightarrow E \rightarrow B \rightarrow F \rightarrow G \rightarrow C \rightarrow A$$

Possible ways of traversing

$3! = 6$



Order: LVR, RVL, LRV, RLV, VRL, VLR

Depth of the node: The depth of the node is number of edges from that node to the root node.

- height of node is the largest no of edges in a path from that node to the leaf node.

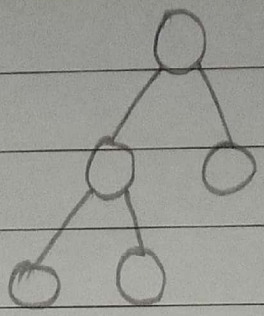
Complete Binary Tree:

A binary tree is a complete binary tree if all the levels are completly filled except possibly the last level and the last level has all the keys as left as

Scanned with CamScanner

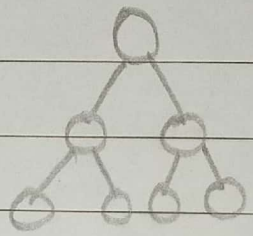
Date

possible.



Perfect Binary Tree:

A perfect binary tree is a binary tree in which all the internal nodes have two children and all nodes are at the same level.



Binary Search Tree:

Binary search tree is a tree in which all the nodes follow the below mention properties

- The value of the left subtree is less than the value of its parent node.
- The value of the right subtree is greater than the value of its parent node

Left subtree $\leq$ parent node $\leq$ right subtree

$$d = \log_2(n+1) - 1$$

Scanned with CamScanner

Date _____

Why AVL?

In some cases we observed that BST's worst case performance is closest to linear search that is O(n). So a need arises to balance out the BST.

AVL:

AVL can be defined as height balanced binary search tree in which each node is associated with a balance factor which is calculated by subtracting the height of its right subtree from its left subtree.

- Tree is said to be balanced if balance factor of each node is b/w -1 to 1, otherwise the tree is unbalance and need to be balanced.

balance factor = height left - height right

Scanned with CamScanner

Date

Hashing:

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.

Hash Table: It is a DS used for storing data.

Collision:

Collision occurs when more than one keys (Hash function) map to the same location of hashes.

Hash Function:

A hash function ussually means a function that compresses meaning the output is shorter than the input.

Types Of Collision Resolution

Open Hashing:

In open hashing collisions are stored outside the table
→ Seprate chaining

Close Hashig:

In close hashing collisions result in storing one of the records at another slot in the table.

Scanned with CamScanner

Date

Types Of Close Hashing:

→ Linear probing
→ Quadratic probing
→ Double Hashing

Hash Function is considered Good if

Fast to compute
Less collisions
Scatter keys evenly throughout the hash table

Applications Of Hashing:

- Hashing is used in password varification and security systems
- It is also used in linking file name and path together
- It is used in comparing complex values
- Compilers use hash tables to implement the symbol table.

Scanned with CamScanner