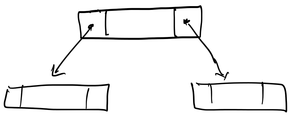
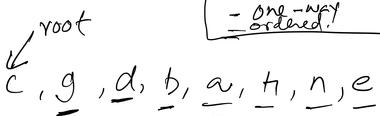
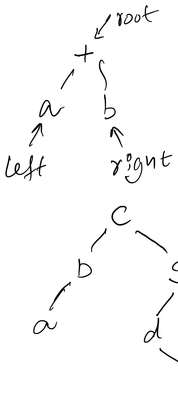
Binary Tree

Sunday, 27 February 2022 10:00 AM



- Dynamic Data Structure
- Pointer
- Non-Adjacent nodes.
- Shrink/grow on the go.
- Size is unknown.
- One-way
- Ordered.





	e.g	Traverse read	write
infix	a+b	inorder	left, root, right
prefix	+ab	preorder	root, left, right
postfix	ab+	postorder	left, right, root

RPN

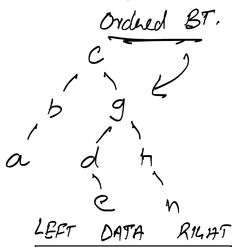
---

c, g, d, b, a, h, n, e

```
//UDT
Record BTNode
    DECLARE LEFT, RIGHT : INT
    DECLARE DATA : CHAR
END Record.

// BT ARRAY DEC.
DECLARE BT : ARRAY [0:9] OF BTNode

// Traverse the BT "INORDER"

PROCEDURE TraverseIO(Root : INT)
IF BT[Root].LEFT <> -1 THEN
    CALL TraverseIO(BT[Root].LEFT)
ENDIF
OUTPUT BT[Root].DATA
IF BT[Root].RIGHT <> -1 THEN
    CALL TraverseIO(BT[Root].RIGHT)
ENDIF
END PROCEDURE.
```

LEFT Root RIGHT

Ordered BT.



	LEFT	DATA	RIGHT	
0	3	c	1	← Root
1	2	g	5	
2	-1	d	7	
3	4	b	-1	
4	-1	a	-1	
5	-1	h	6	
6	-1	n	-1	
7	-1	e	-1	
8	-1		-1	
9	-1		-1	

BT.

CALL #	PARAM	OUTPUT	STATUS
1	0		SUSPEND
2	3		SUSPENDED
3	4	a	FINISHED
2	3	b	RESUME
1	0	c	FINISHED RESUME
4	1		SUSPEND SUSPENDED
5	2	d	SUSPEND
6	7	e	Finish
5	2		Finished
4	1	g	Resume

b → Root

Left ← a c → Right

Inorder: a b c Sorted (ASC).

Preorder: b a c

Postorder: a c b (RPN)

```
// Insert Into BT.

PROCEDURE INSERT_BT(NodeData : CHAR).
    IF BTSize = 10 THEN
        OUTPUT "Overflow ____"
    ELSEIF BTSize = 0 THEN
        BT[0].DATA = NodeData
        BTSize = 1
    ELSE
        CN = Root    // N=0
        REPEAT
            PN = CN
            IF NodeData < BT[CN].DATA
            THEN
                Direction = 'L'
                CN = BT[CN].LEFT
            ELSE
                Direction = 'R'
                CN = BT[CN].RIGHT
            ENDIF
        UNTIL BT[CN].DATA = ''

        BT[CN].DATA = NodeDAT
        IF DIRECTION = 'L' THEN   BT[PN].LEFT = CN
        IF DIRECTION = 'R' THEN   BT[PN].RIGHT = CN
        BTSize = BTSize + 1
    ENDIF
END PROCEDURE.
```