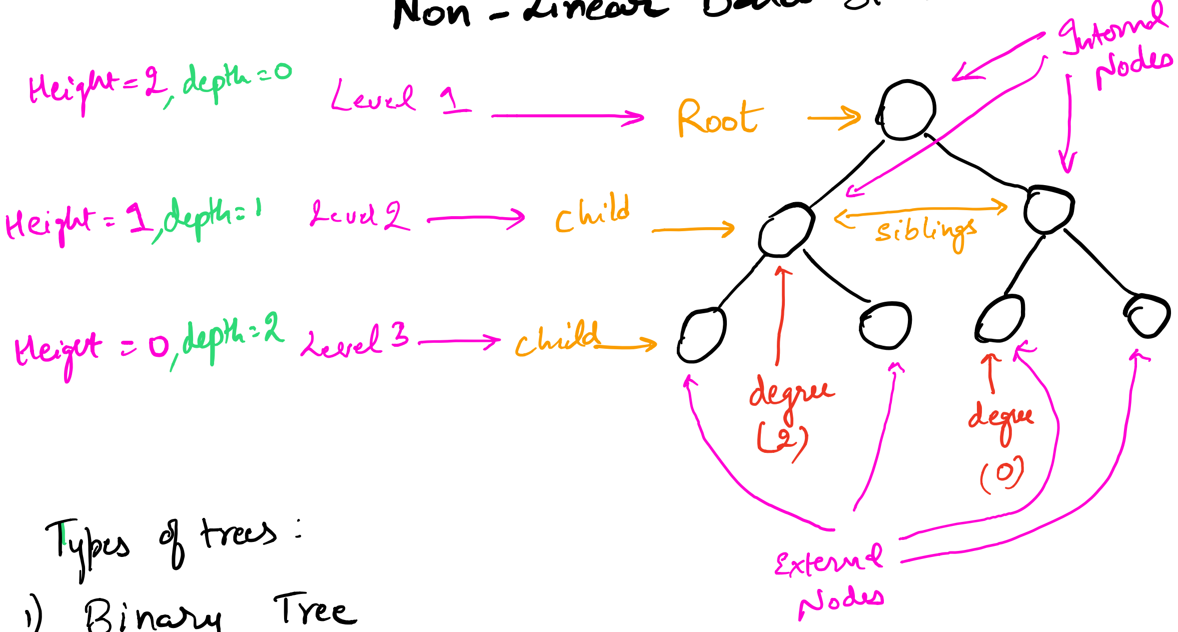
Trees

↓

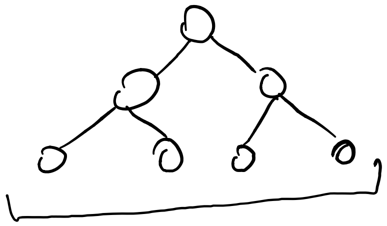
Non - Linear Data Structure

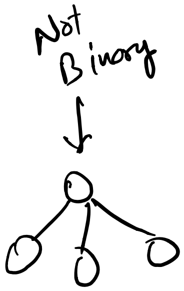


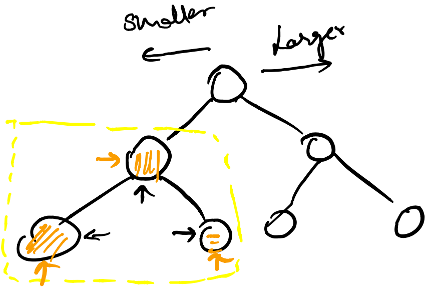
Types of trees :

1) Binary Tree

2) Binary Search Tree





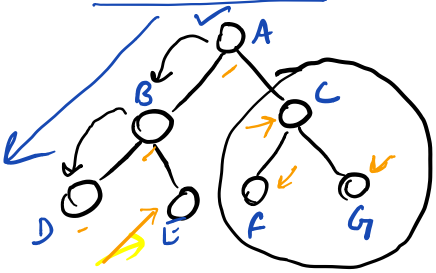


TREE TRAVERSAL

1) Depth First Search (DFS)

2) Breadth First Search (BFS) (Levelling Search)

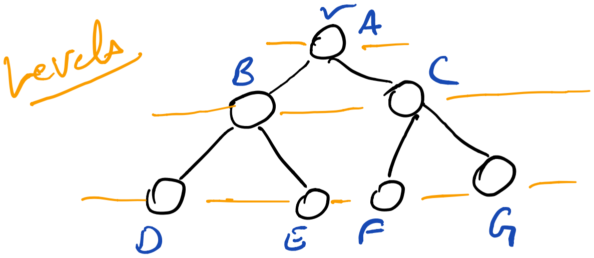
Depth First Search



A B D E C F G

(STACKS)

Breadth First Search



A B C D E F G

(QUEUES)

Depth First Search

Types:

1) INORDER
   - Traverse left subtree
   - visit Root Node
   - Traverse Right subtree

   [DBEAFCG]

2) PREORDER
   - Visit Root Node ✓
   - Traverse left subtree ✓
   - Traverse Right subtree ✓

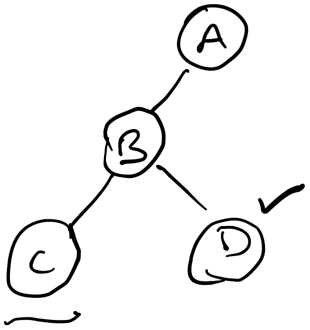
   [ABDECFG]

3) POST ORDER
   - L — Traverse left subtree
   - Ri — Traverse Right subtree
   - Root — Root (Visit)

   [DEBFGCA]

Algorithm : INORDER (ROOT)

```
1.  P = ROOT
2.  Initialize Stack
3.  Repeat while Stack is not empty or P ≠ NULL
4.        Repeat while P ≠ NULL
             a)  PUSH (Stack, P)
             b)  P = P → LCHILD
          [End of loop]
```



```
5.  If Stack is not empty then
          a)  P = POP(Stack)
          b)  Print : P → Info (data)
          c)  P = P → RCHILD
       [End of loop]
6.  Return
```