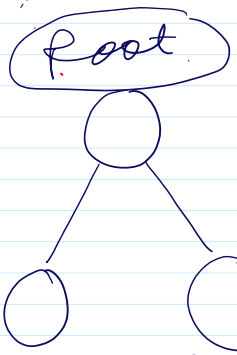
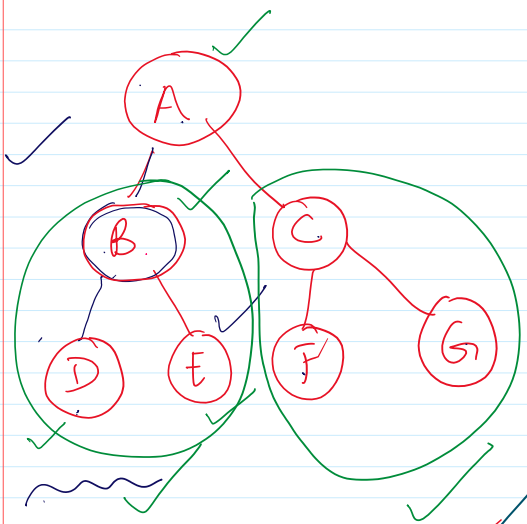
27 June 2025 01:50 PM

Tree and Graph Traversals.

1) Preorder (Root Left Right)
2) Postorder (L R Root)
3) In-order (L Root R).



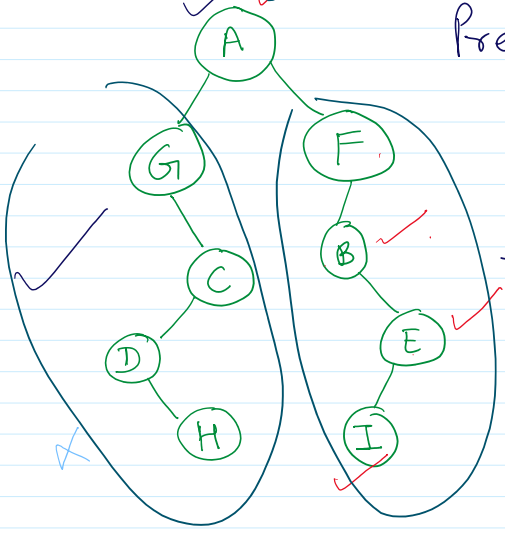


Preorder → A B D E C F G

Postorder → D E B F G C A

Inorder → D B E A F C G
(L R L R)

Q-2



Preorder

A G C D H F B E I

Inorder (Left Root Right)

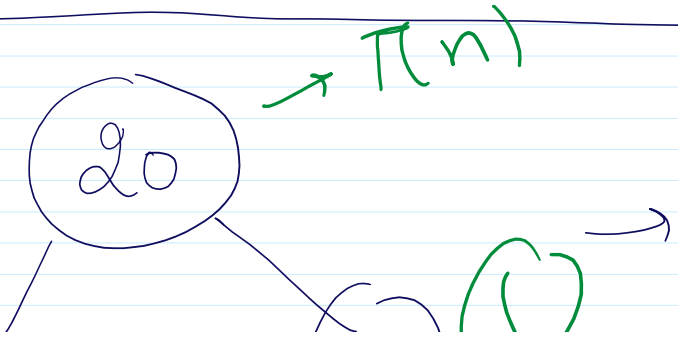
G D H C A B I E F

(Root)

Postorder →

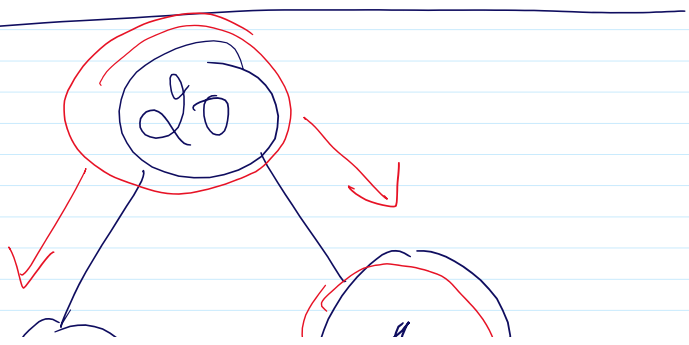
(Left Right Root) → H D C

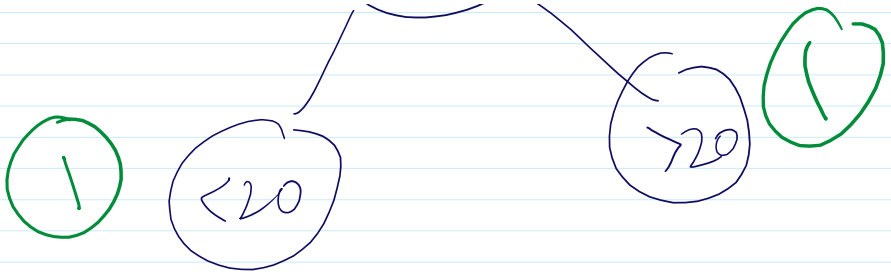
BST



*)

GIEBFA





Inorder of BST → Ascending order

O(n) Preorder (root) → T(n) Root L

{ Print (root → data) → O(

Preorder (root → left chil

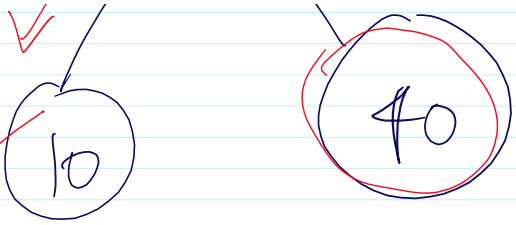
Preorder (root → right ch

} $T(n) = T(n/2) + T(n/2$

$T(n) = 2T(n/2) + C$

Master's Theorem

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$



10	20	40

eft Right

1)

d) → T(n/2)

uild) → T(n/2)

2) + O(1) → C

$$T(n) = 2T\left(\frac{n}{2}\right) + C$$

$n^{\log_b a}$ Compare $f(n)$

$$n^{\log 2^2} = n^1 = n$$

$f(n) = C$, n

Postorder (root) → $T(n)$

{ Postorder (root → lc) → T

Postorder (root → rc) →

Print (root → data) → (

}

$$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$$

$a = 2$

$b = 2$

$f(n) = C$

$T(n) = O(n)$.

$(n|2)$

$T(n|2)$

$O(1)$

C

$T(n) = O(n)$

Conclusion : Preorder, Inorde

$O(n)$ ✓

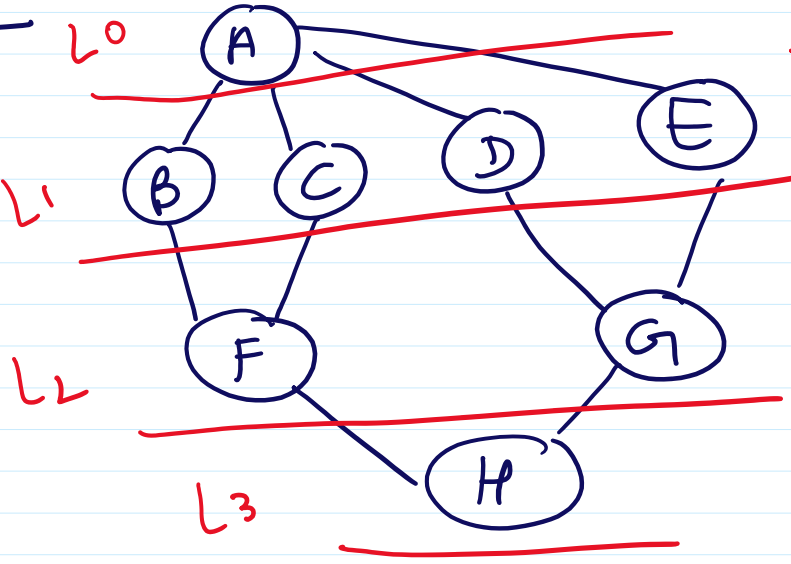
ee, Post order

Graph Traversals

BFT (Breadth First Search

DFT (Depth " ")

BFT



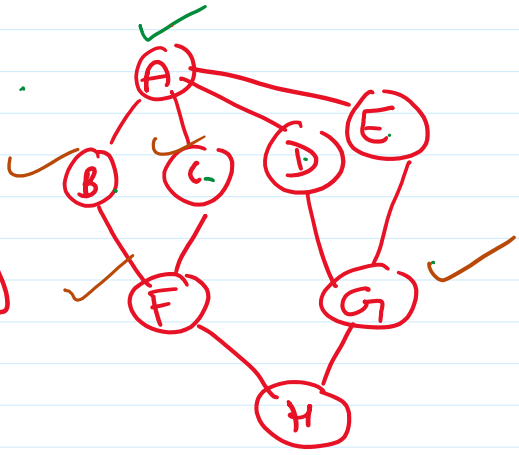
A B C D E F G H

Level order Traversal

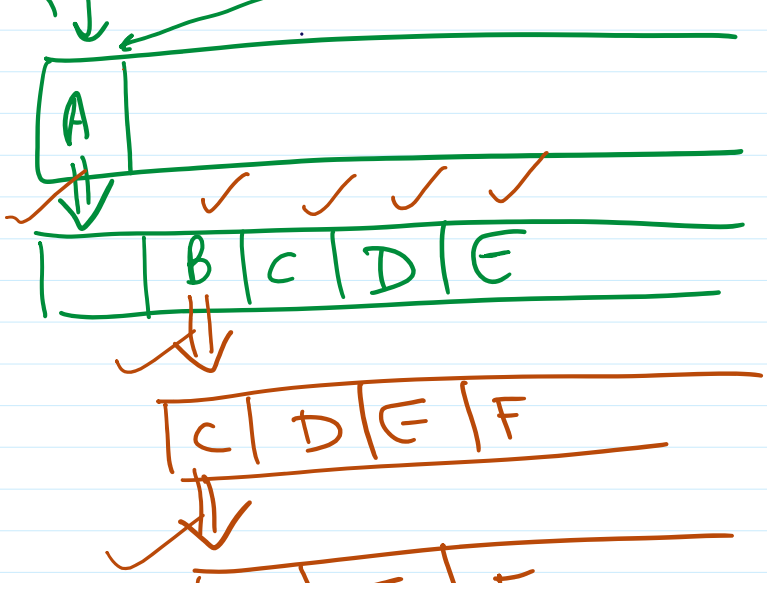
BFT(V)

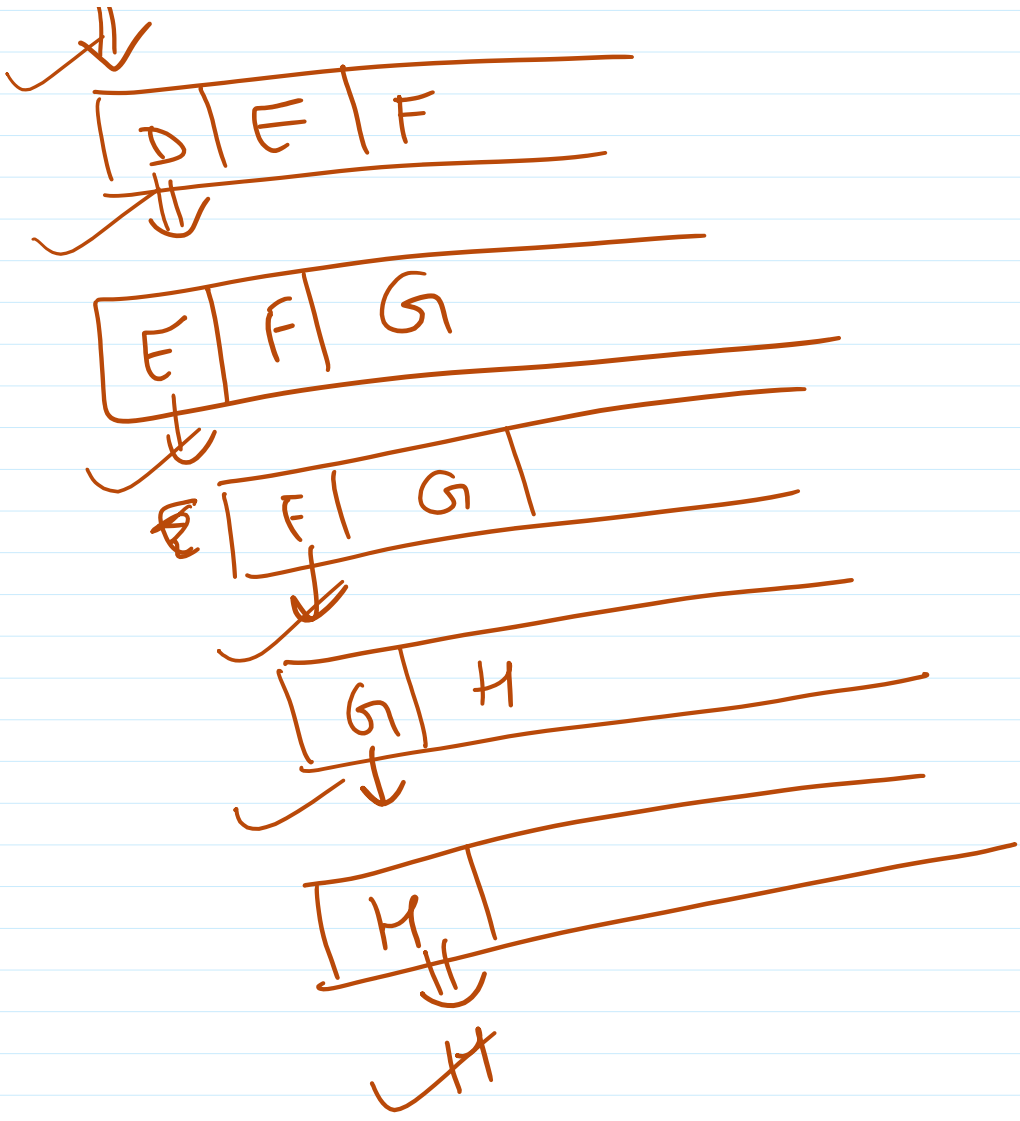
visited = 0 0 0 0 0 0 0 0

A B C D E F G H



Queue

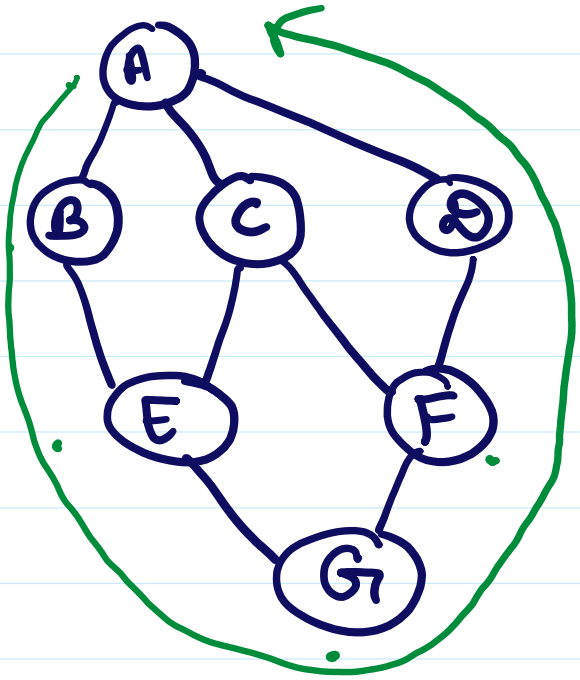





```
BFT(v)
{ visited(v) = 1 ✓
  add(v, Q) ✓
  while (Q is not empty) ✓
  { x = delete(Q) ✓
    Print(x) ✓
    for all w adjacent to x
    { if w is not visited)
```

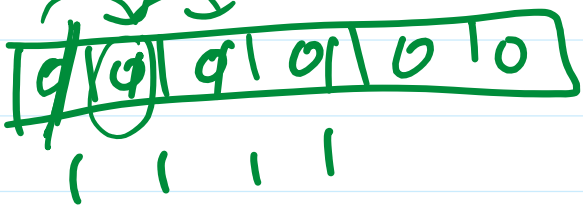
{ if w is
{ visited(w) = 1
} } } add(w, Q)

DFT



A B E G F C D

visited

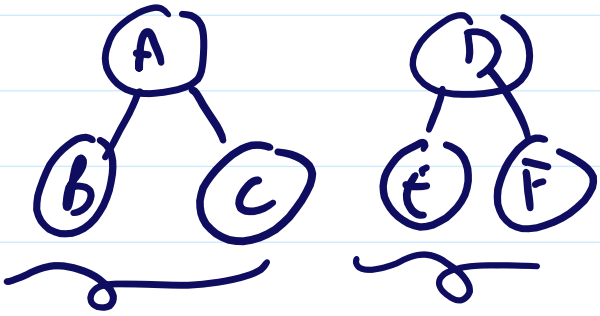


```
DFT(v)
{ visited(v)= 1
  Print(v)
for all w adj to v
    if (w is not visited)
      DFT(w)
  }
}
```

BFT

1) Given a graph is connected or not

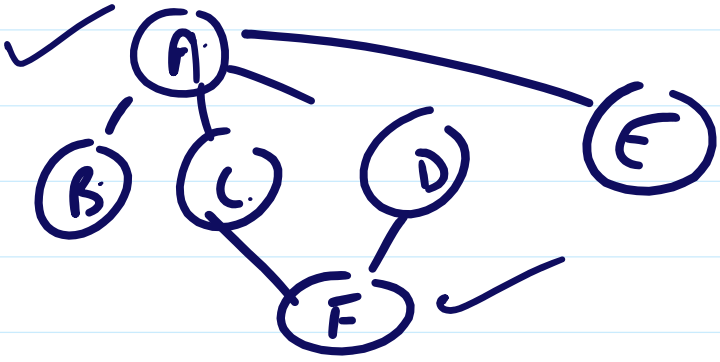
1) Given a graph is connected or not)
(BFT, DFT)



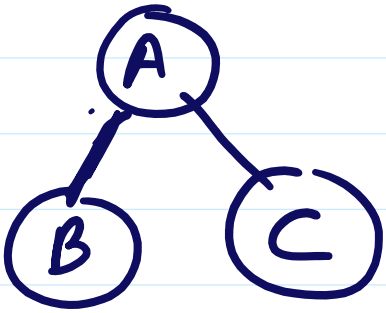
2) Connected
(BFT, DFT)

3) Graph is Cyclic or not)

4) <u>Using BFT → Shortest Path</u>



Adj matrix



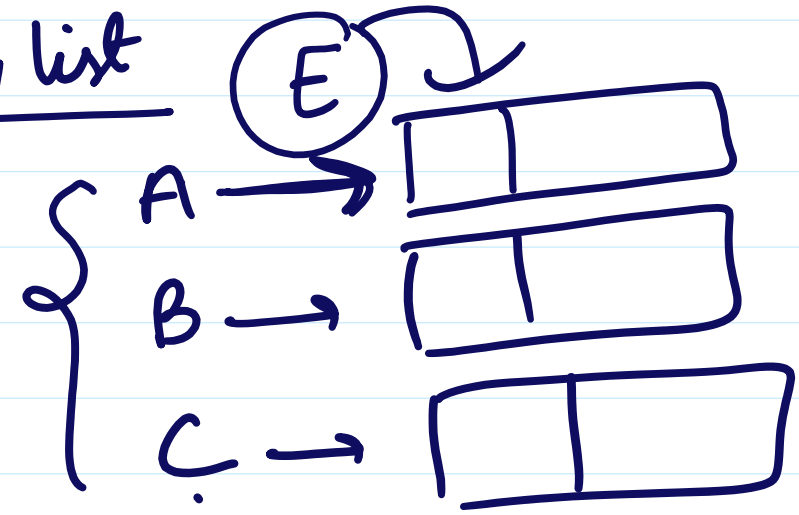
	A	B	C
A	ϕ	1	1
B	1	0	0
C	1	0	1

→ $V \times V = V^2$

$O(V^2)$

Adjacency list

Adjacency list



O(V)

O(V+E)

Matrix → O(V²)

Adj list → O(V+E) ✓

✓ ✓ ✓ } BFT DFT } Queue.