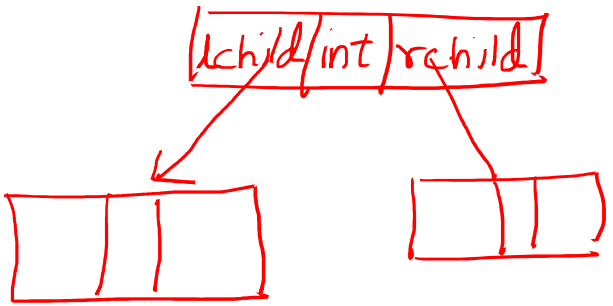
Binary Tree : Traversal Techniques

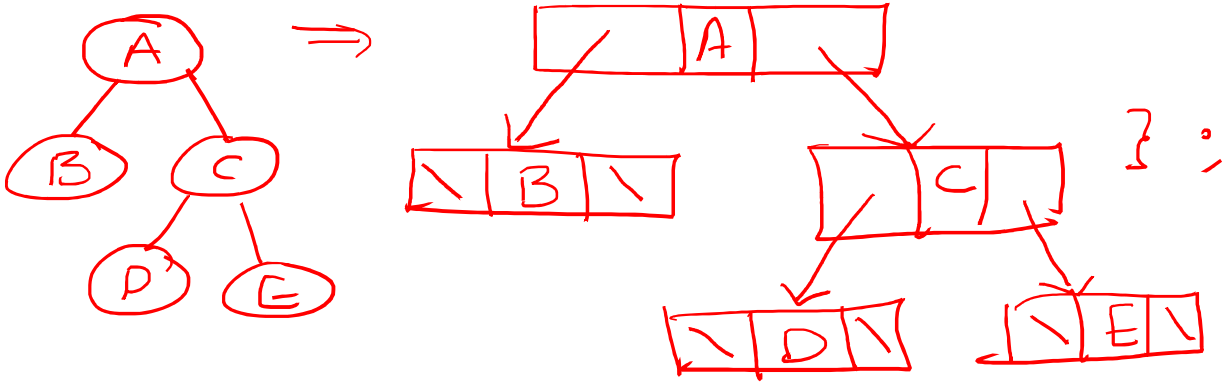
Dec. 04th, 2021

Lecture-18

(2) linked list representation of BT

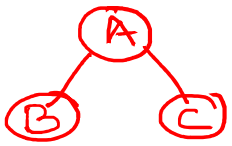



```
class btree{ int data;

btree *lchild;
btree *rchild;

public;
=
=
};
```



Traversal Techniques:

- Inorder traversal → LVR
- Postorder traversal → LRV
- Preorder Traversal → VLR
- Level-order traversal → level wise display (L → R)

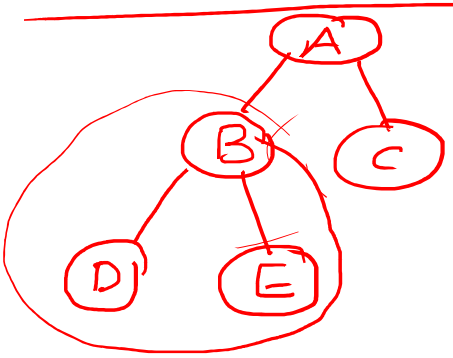


Inorder = B A C

postorder = B C A

preorder = A B C

levelorder = A B C

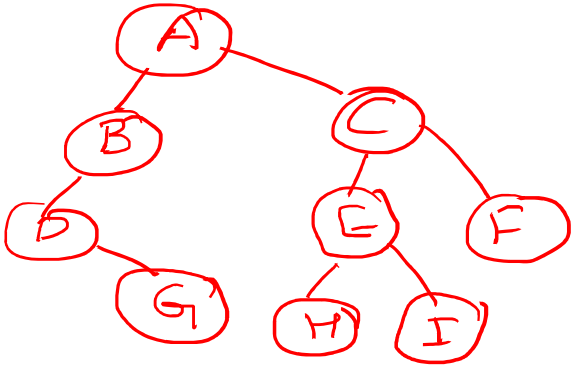


In: D B E A C

post: D E B C A

pre: A B D E C

level: A B C D E

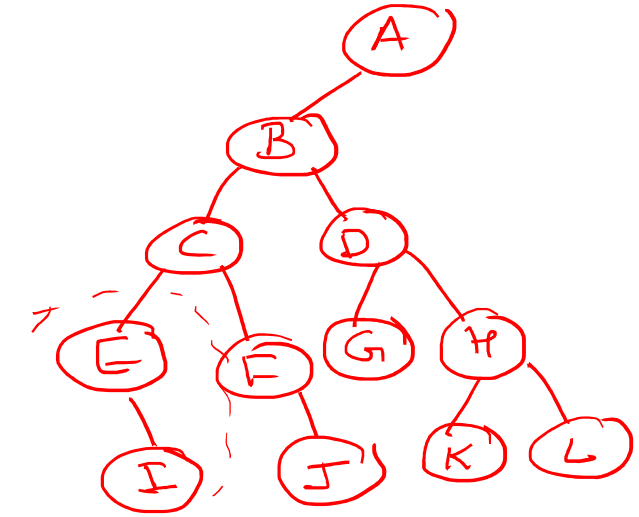


In: D G B A H E I C F

post: G D B H I E F C A

pre: A B D G C E H I F

level: A B C D E F G H I



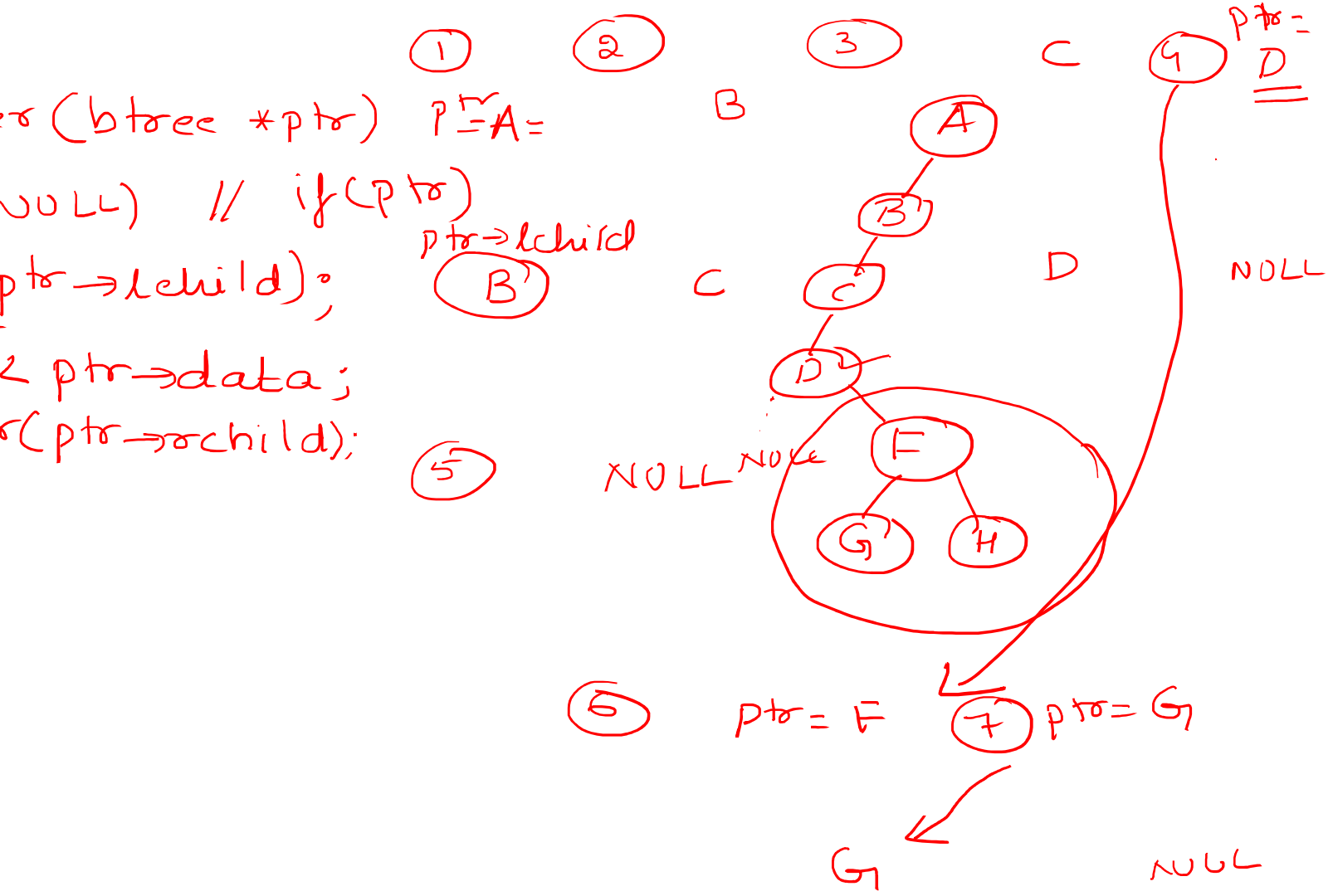
In: E I C F J B G D K H L A

post: I E J F C G K L H D B A

pre: A B C E I F J D G H K L

level: A B C D E F G H I J K L

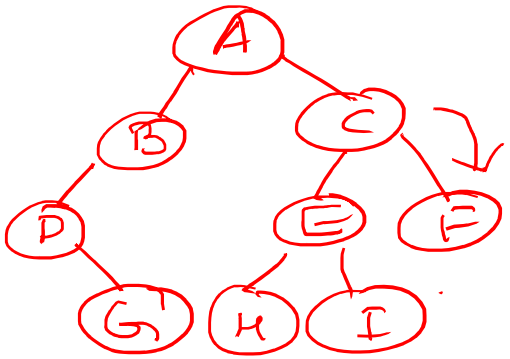
```
void inorder(btree *ptr)
{
    if (ptr != NULL)
    {
        inorder(ptr->lchild);
        cout << ptr->data;
        inorder(ptr->rchild);
    }
}
```



D G

Recursive inorder

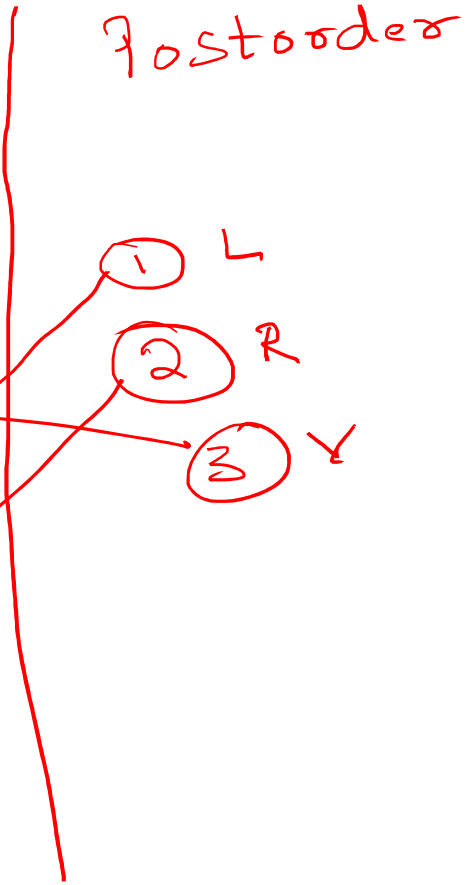
fn. call	ptr	ptr→child
1	A	
2	B	
3	D	cout << p...
4	NULL	
5	G	cout << G.
6	NULL	
7	NULL	
8	C	
9	E	
10	H	Rout << H
11	NULL	



NULL

Preorder

```
void preorder(btree *ptr)
{   if(ptr!=NULL)
    { cout<<ptr->data;
      preorder(ptr->lchild);
      preorder(ptr->rchild);
    }
}
```

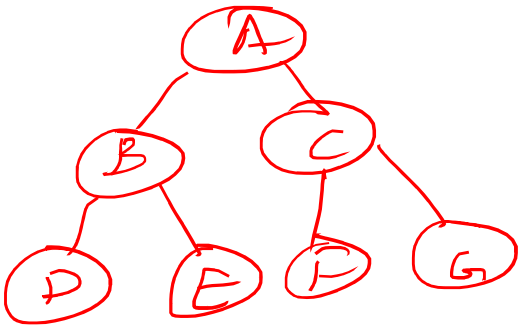


Level-order

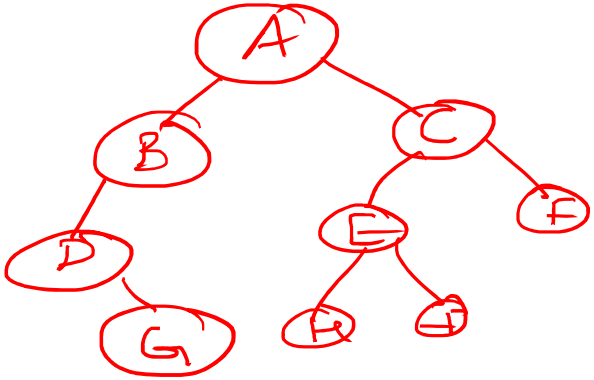
0 1 2

A B C D E F G

Queue



A B C



A B C D E F G H I

A B C D E F G H I