

root \rightarrow

children \rightarrow

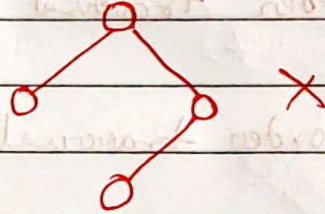
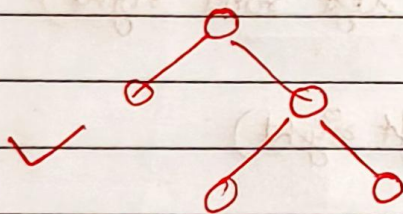
leaf node \rightarrow

ancestors \rightarrow

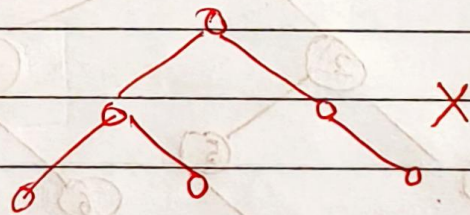
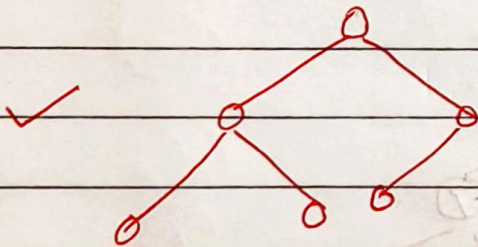
Subtree \rightarrow

Descendants \rightarrow

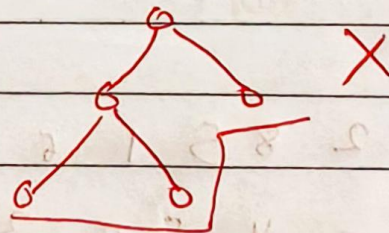
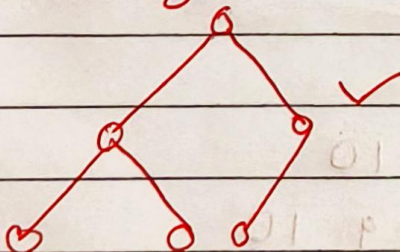
Full BT \rightarrow either has 0 or 2 children



Complete BT \rightarrow All levels are completely filled except the last level. The last level has nodes in left as possible

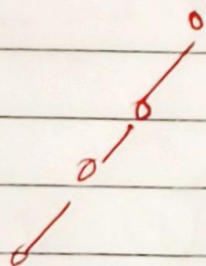


Perfect Binary tree: All leaf nodes are on same level.



Balanced tree \rightarrow Max height can be maximum of $\log(n)$

Degenerated tree \rightarrow



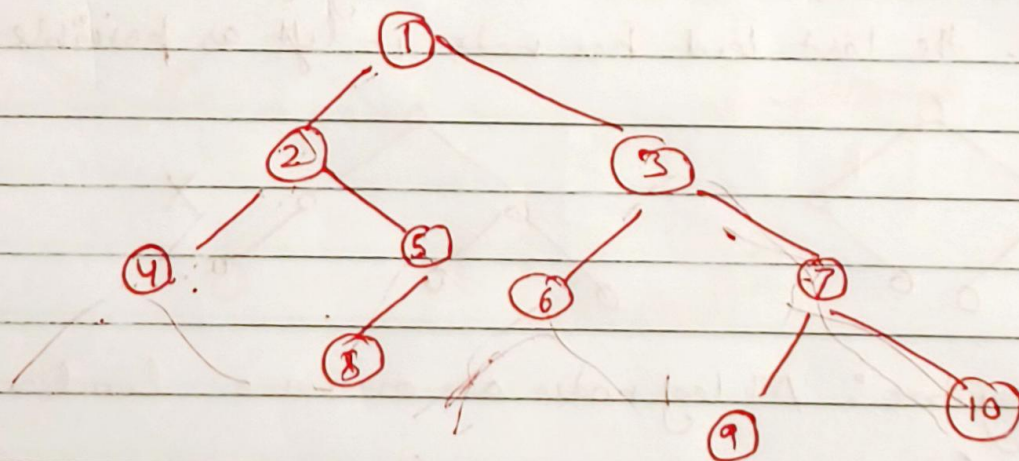
every node has single children.

DFS

\rightarrow Inorder traversal (~~Root~~ left root right)

\rightarrow Pre-order traversal (root left right)

\rightarrow Post-order traversal (left right root)



4 2 8 5 1 6 3 9 7 10

1 2 4 5 8 3 6 7 9 10

4 8 5 2 6 9 10 7 3 1

BFS

1 2 3 4 5 6 7 8 9 10

Representation

```
struct Node {
```

~~Node~~

```
    int data
```

```
    struct Node* left;
```

```
    struct Node* right;
```

```
Node (int val)
```

```
{
```

```
    data = val;
```

```
    left = right = NULL;
```

```
}
```

```
main()
```

```
{
```

```
    struct Node* root = new Node(1);
```

```
    root → left = new Node(2);
```

```
    root → left = new Node(3);
```

```
    root → left → right = new Node(5);
```

```
}
```