

<https://my.newtonschool.co/playground/code/ev8574juvilj>

Newton School

×

Newton School

×

Newton School

×

+

→

↺

my.newtonschool.co/playground/code/ev8574juvilj

Java (OpenJDK 15.0.1)

🔍

🌟

🔧

🔄

📄

👤

⋮

Run

Save

```

1 static Queue<Integer> ReverseK(Queue<Integer> queue, int k) {
2
3
4
5     Stack<Integer> stack = new Stack<Integer>();
6
7     // Push the first K elements into a Stack
8     for (int i = 0; i < k; i++) {
9         stack.push(queue.peek());
10        queue.remove();
11    }
12
13
14    // Enqueue the contents of stack at the back
15    // of the queue
16    while (!stack.empty()) {
17        queue.add(stack.peek());
18        stack.pop();
19    }
20
21
22    // Remove the remaining elements and enqueue
23    // them at the end of the Queue

```

n

k

5

3

1

2

3

4

5

Stack

1

2

3

4

5

Queue

1

2

3

4

5

for (int i = 0; i < k; i++) {

stack.push(queue.peek());

queue.remove();

←

1

2

3

4

5

while (!stack.empty()) {

queue.add(stack.peek());

stack.pop();

←

1

2

3

4

5

for (int i = 0; i < queue.size() - k; i++) {

queue.add(queue.remove());

←

1

2

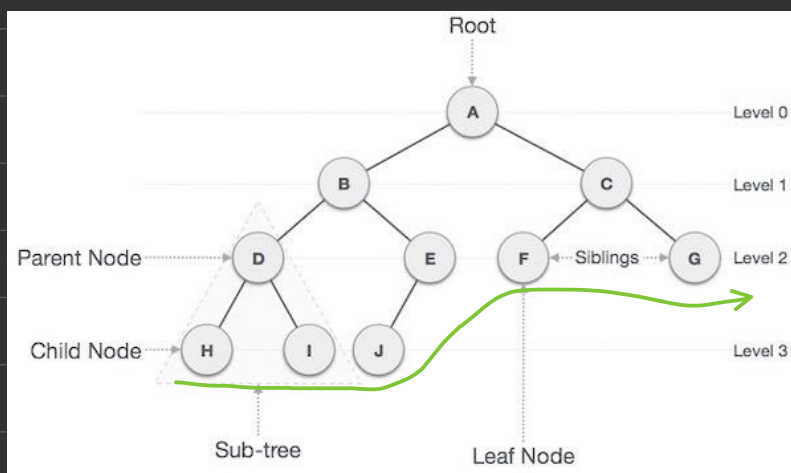
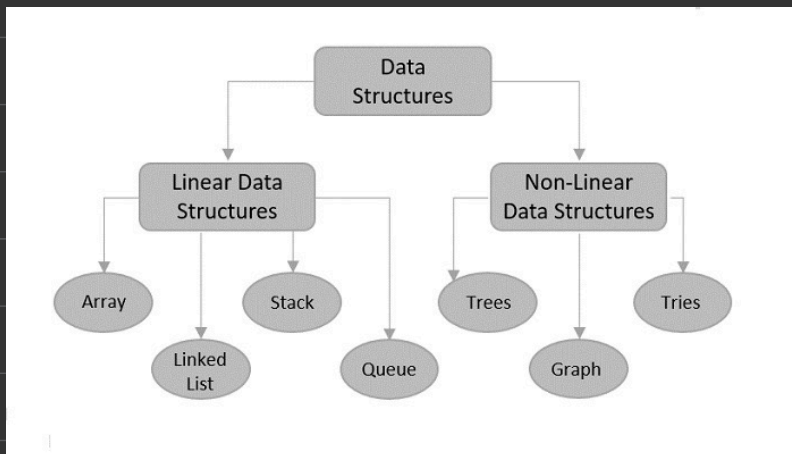
3

4

5

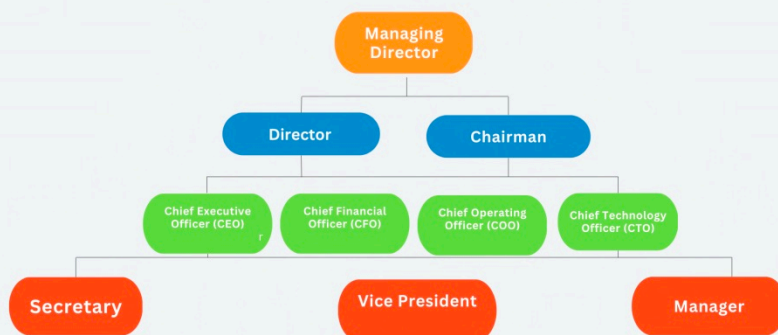
## Trees: Binary Trees

-non linear data structure

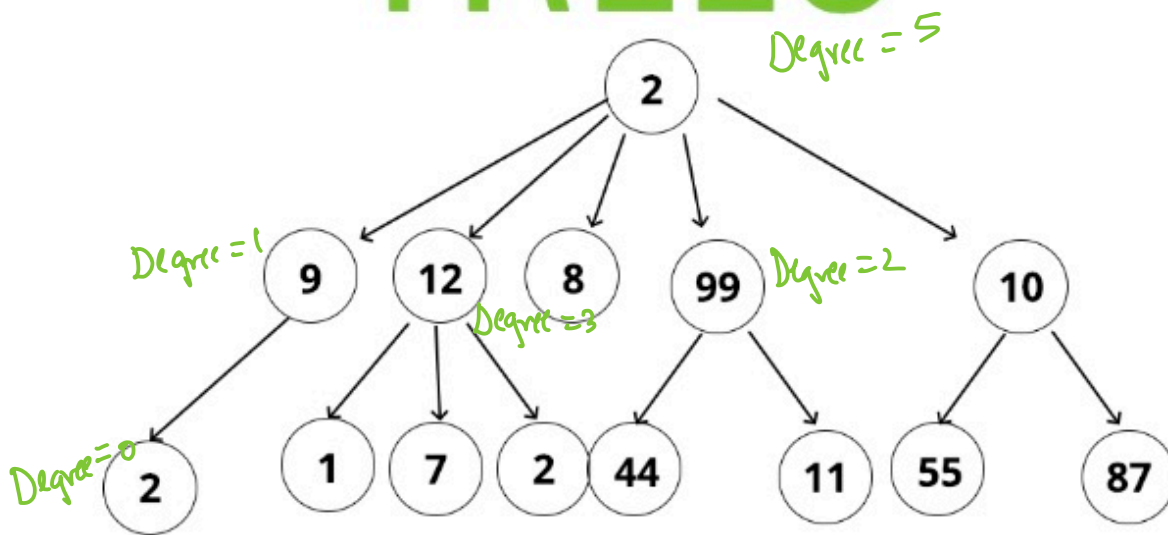


[www.instafling.com](http://www.instafling.com)

## DESIGNATIONS IN A PRIVATE LIMITED COMPANY



# TREES



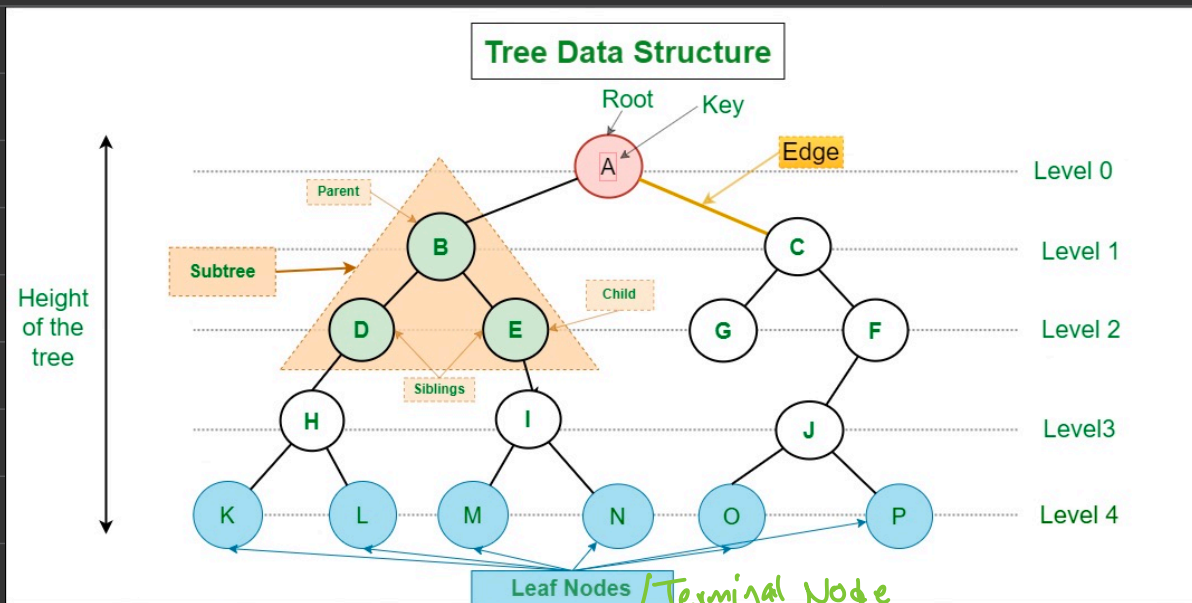
Degree = 0 | 1 | 3 | 5 | etc.

<https://medium.com/codestorm/linear-and-non-linear-data-structure-7f10dc9e9a9f>

[https://www.youtube.com/watch?v=sOUtHG7O\\_2g](https://www.youtube.com/watch?v=sOUtHG7O_2g)

<https://www.youtube.com/watch?v=03DErzKxoX0>

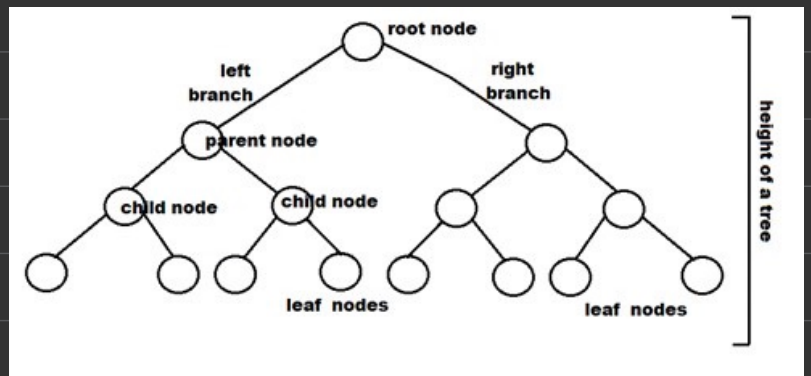
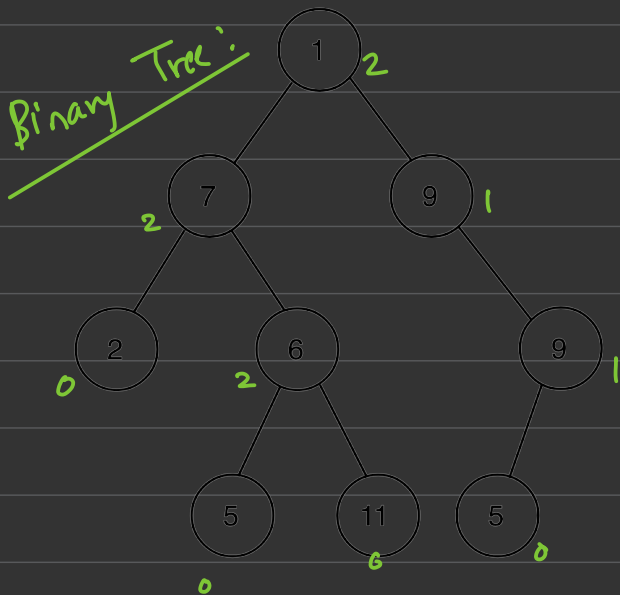
Binary Tree: Degree = 0 or 1 or 2.



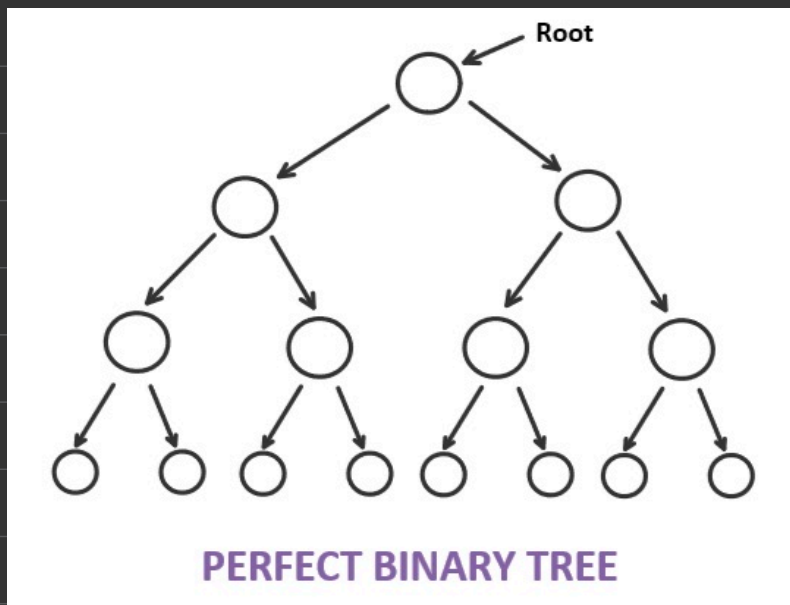
○ — Vertices  
 / — Edges

Height of Tree: The length of the longest path from that node to the terminal node

Degree: Degree of a node is the number of child node it has.



**Complete Binary Tree:** A binary tree is called as a complete binary tree if it contains maximum number of nodes it can have.



Properties of Binary Tree:

- >A binary tree with 'n' nodes has exactly 'n-1' edges.
- >In a binary tree, every node except the root node has exactly one parent.
- >In a binary tree, there is exactly one path connecting any two nodes in a tree.
- >The minimum number of nodes in a binary tree of height 'h' is 'h+1'.
- >The maximum number of nodes in a binary tree of height 'h' is  $2^{h+1} - 1$   
eg: height = 4, #nodes =  $2^{4+1} - 1 = 2^5 - 1 = 32 - 1 = 31$ .
- >Number of leaf nodes in a complete binary tree is  $\frac{n+1}{2}$

>In a complete binary tree,  
Number of external nodes = Number of internal nodes + 1

