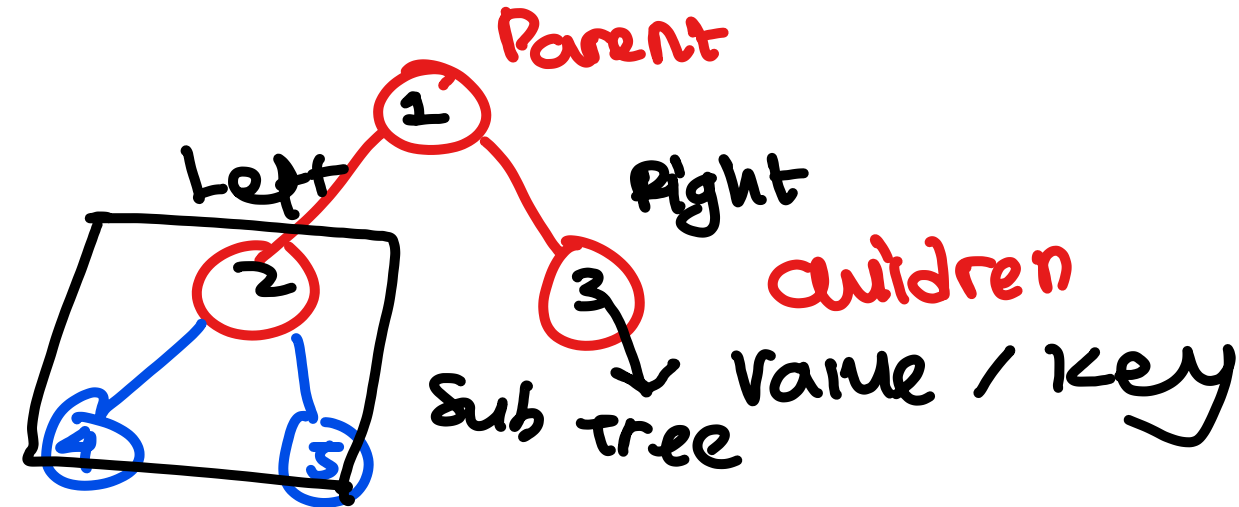


Binary & N-Ary Tree

By Abdan Hafidz

Binary ^{Tree} \Rightarrow 1 Parent 2 child



Tree
Left * tree
Right *

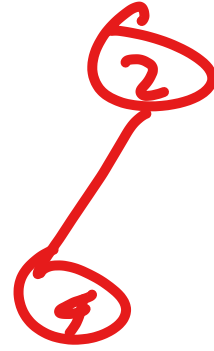
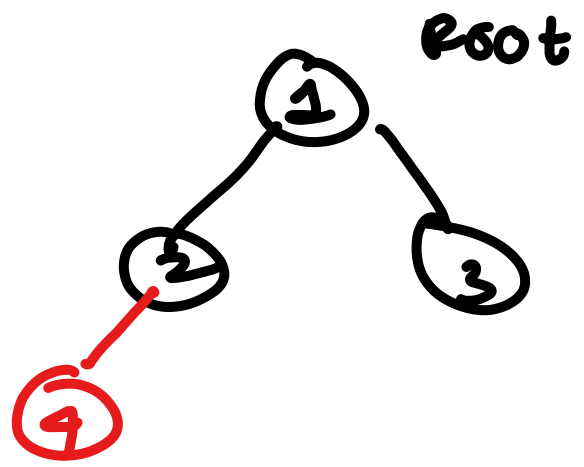
null

String *kata = null ptr

String kata = null

==

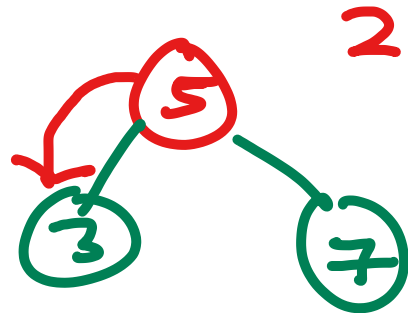
If (kata == "")
If (kata == null) False



Binary Search Tree

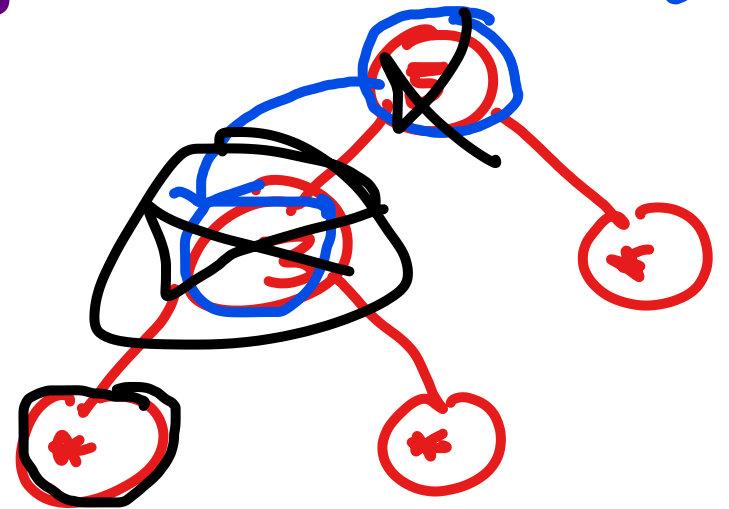
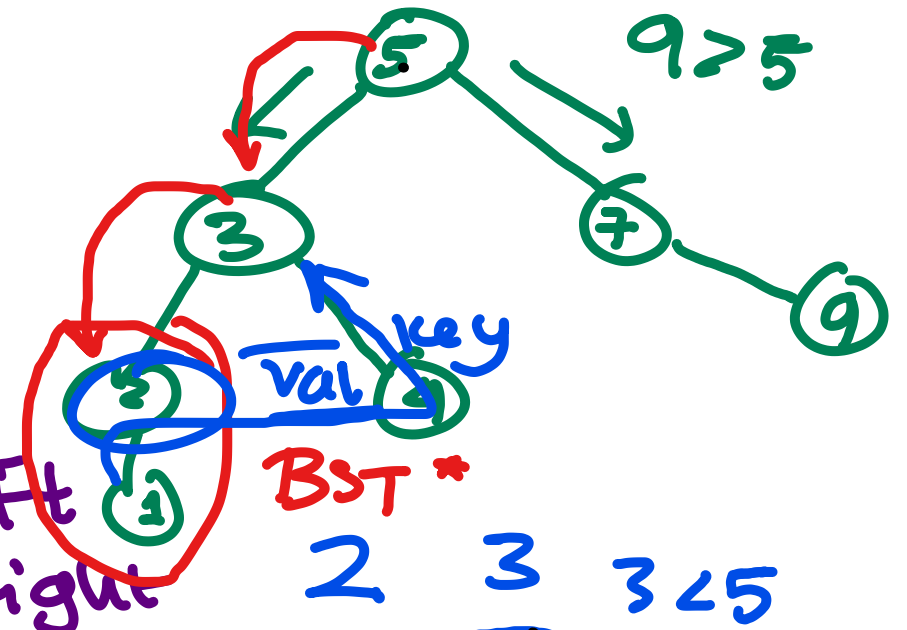
5 3 7 9 4 2 1
 ↓
 Root

current val
 current val

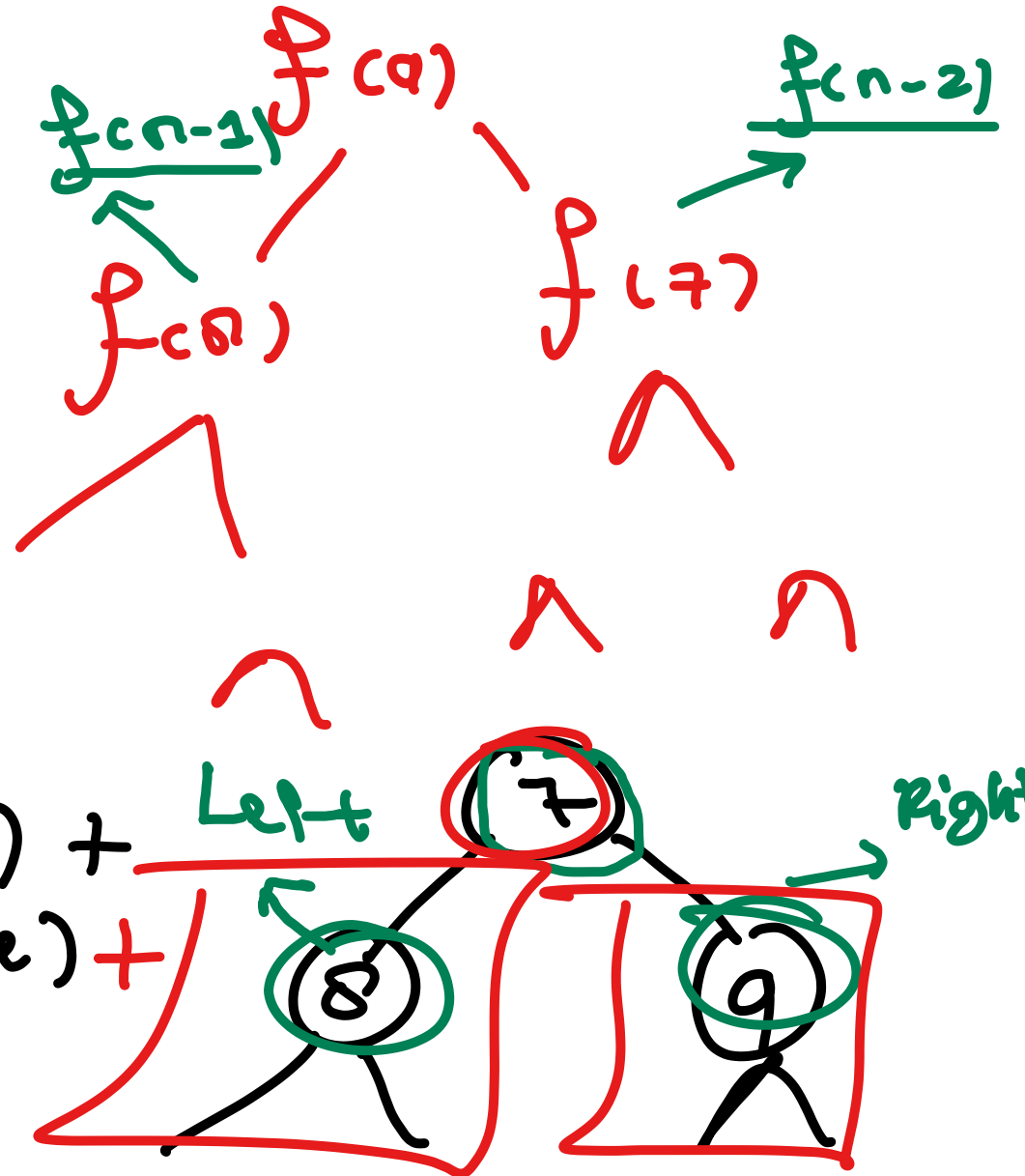


< Root → left
 > Root → right

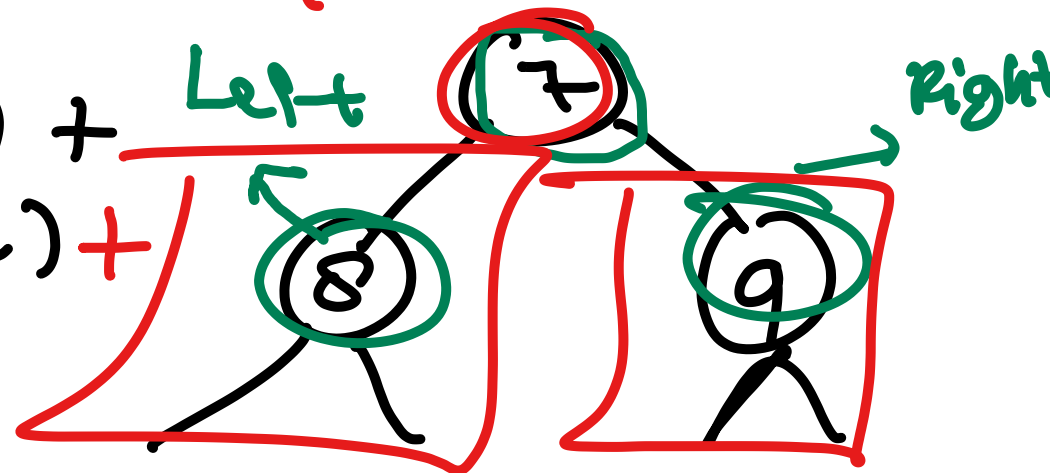
Binary

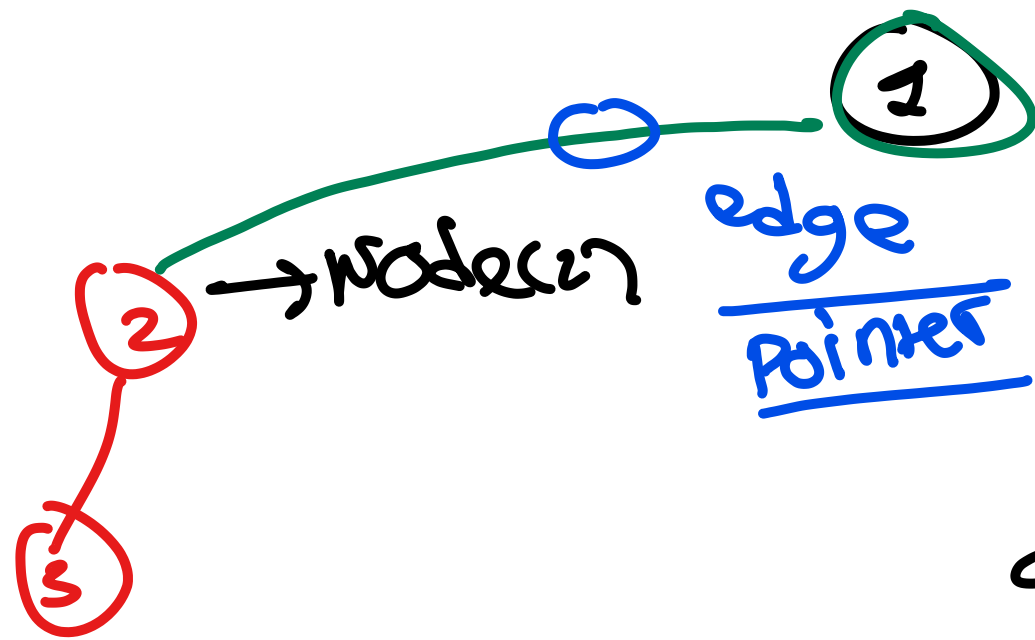


$$\underline{f(n)} = \underline{f(n-1)} + \underline{f(n-2)}$$



$$\text{sum}(\text{node}) = \text{sumLeft}(\text{node}) + \text{sumRight}(\text{node}) + \text{key}$$





Node 1

node (2)

= node (1) → left

node (2) → left = 3

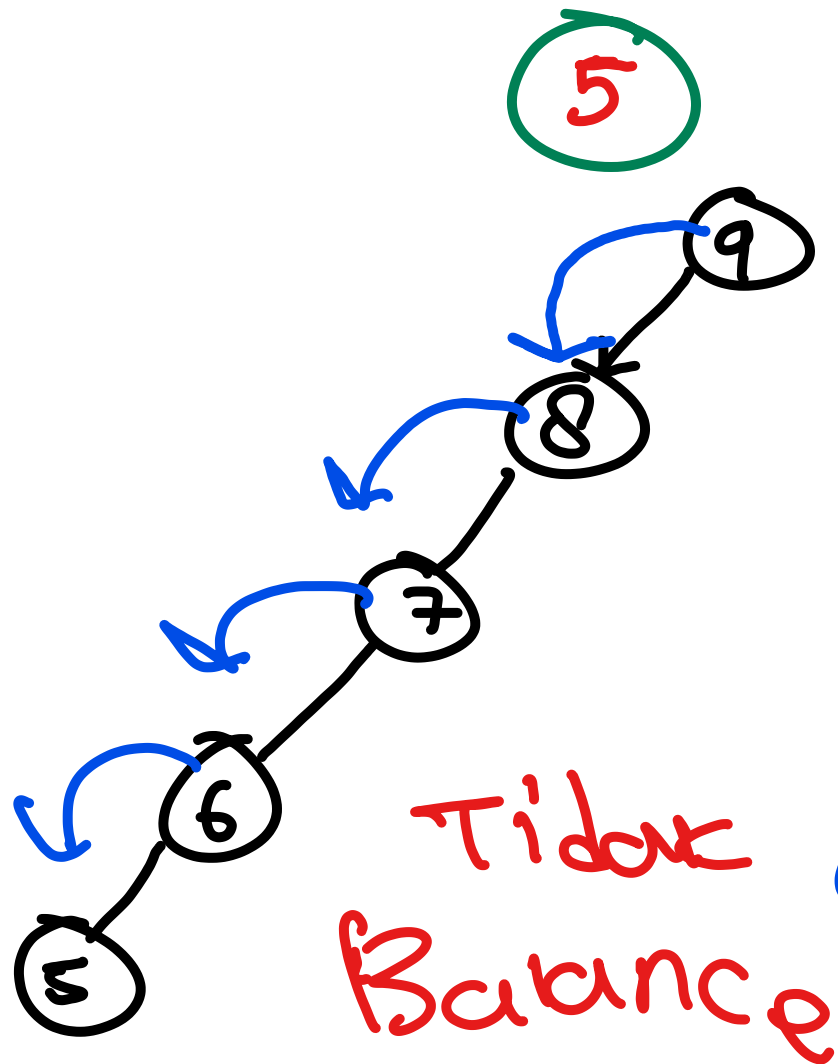
cout << node (2) → left → left = 3

node 2 → left

pointer

node (2) = copy node (1) → left

9 8 7 6 5
↓



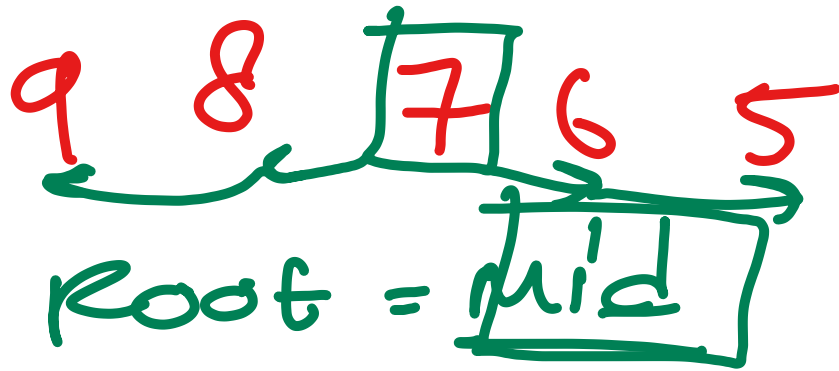
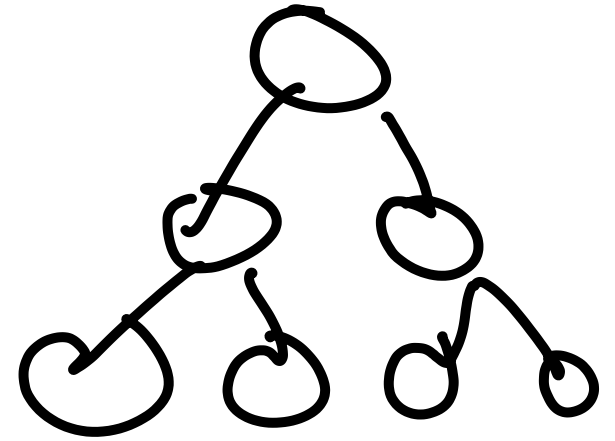
3 langkah

4 langkah

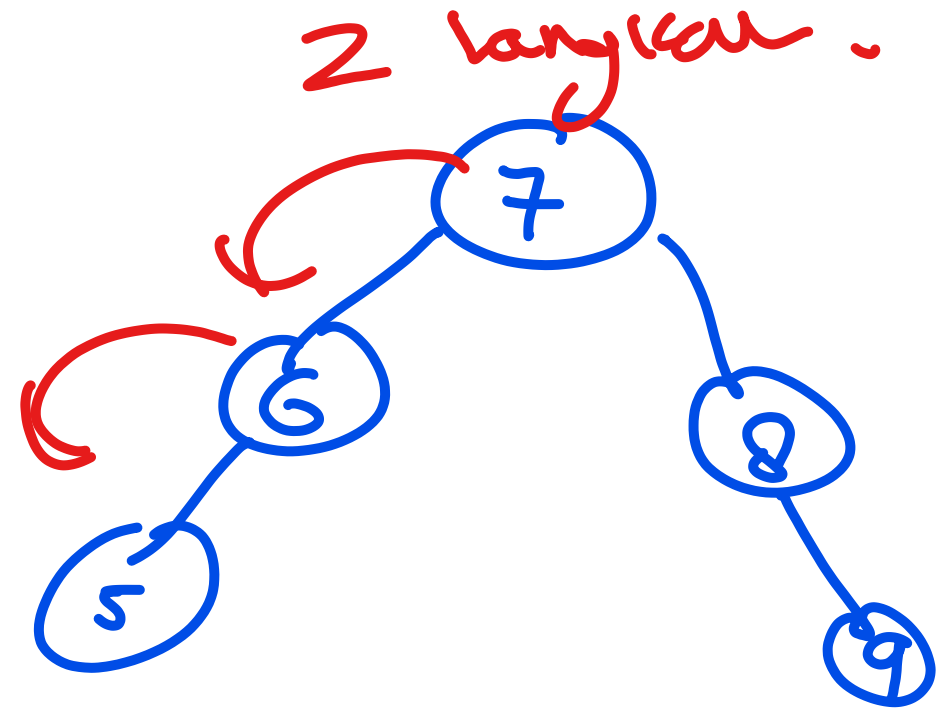
$O(N)$

$O(\log N)$

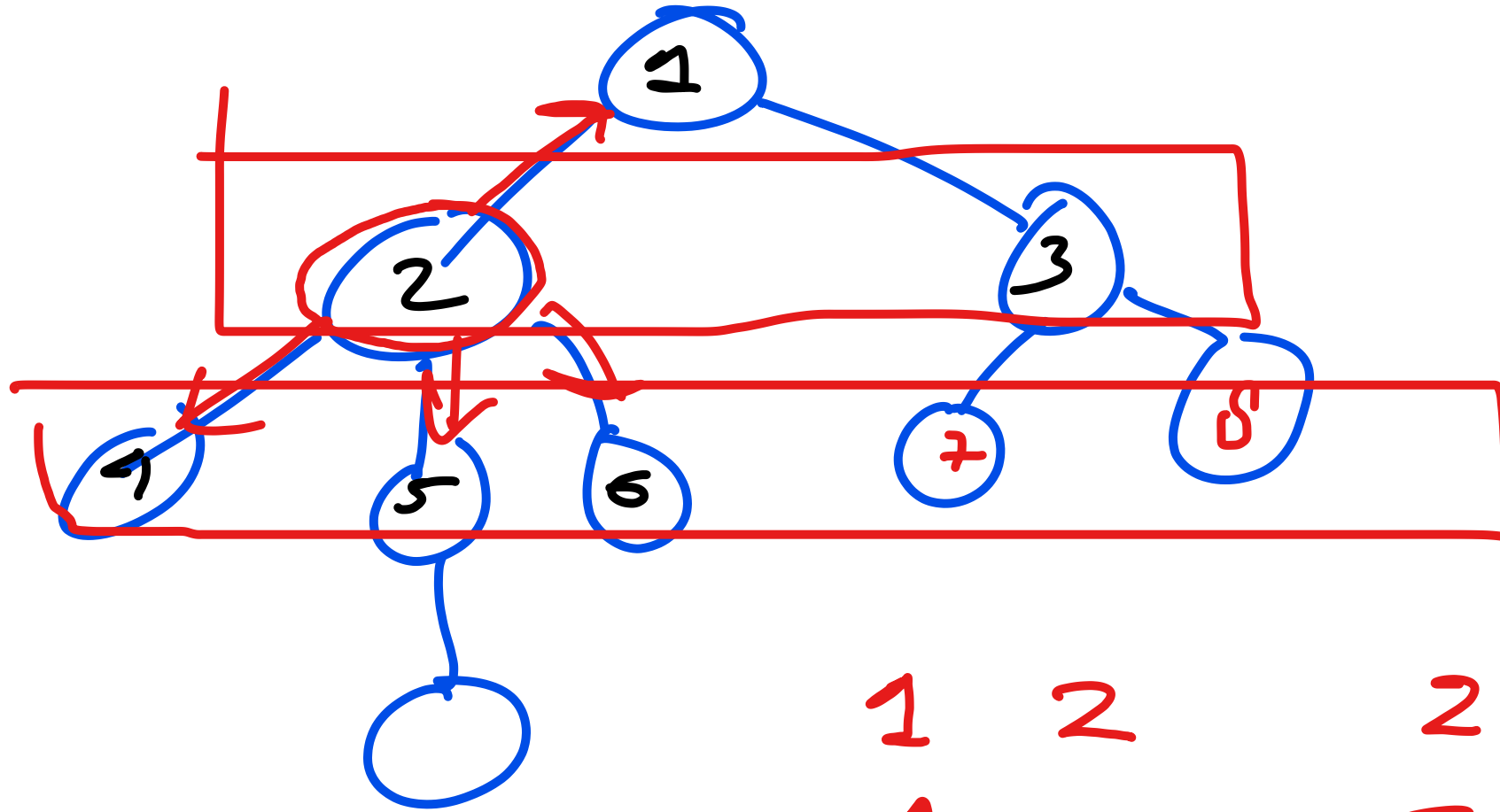
$H_{Left} = H_{Right}$



AVL tree



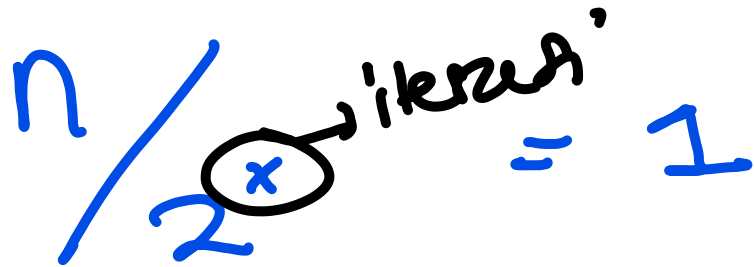
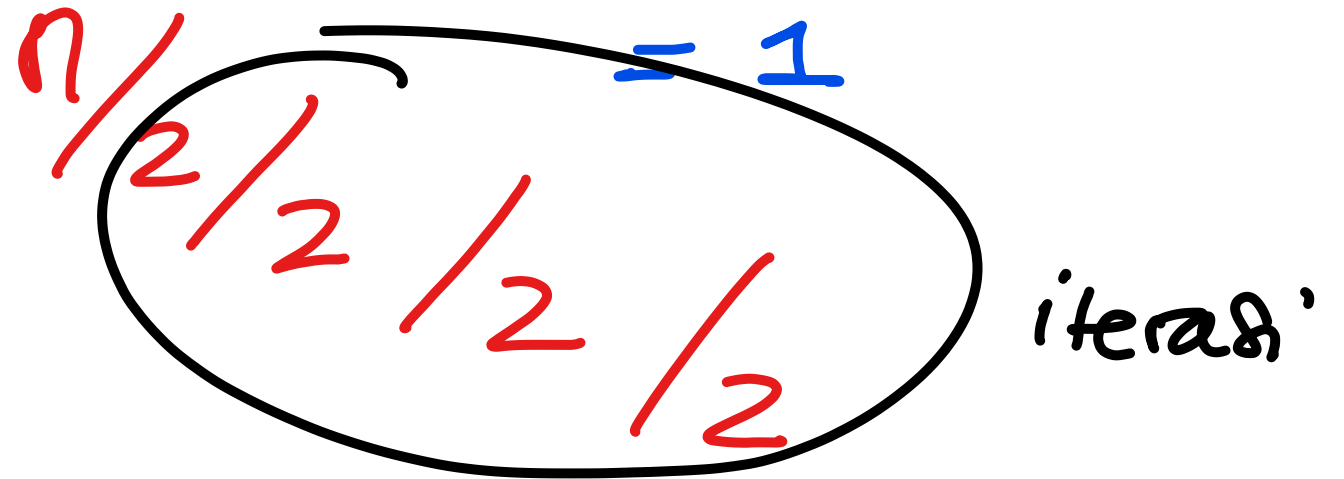
N - Any Tree



1	2	2	4
1	3	2	5
		2	6

```
int n;  
cin>>n;
```

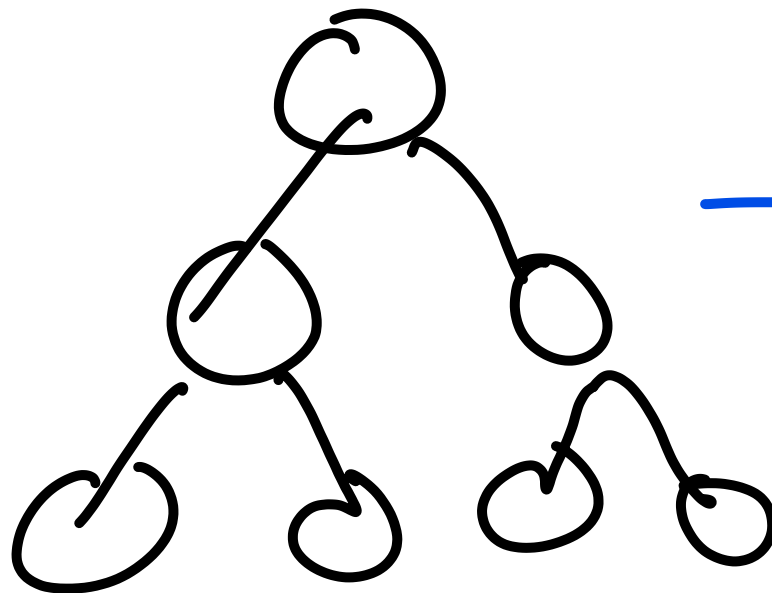
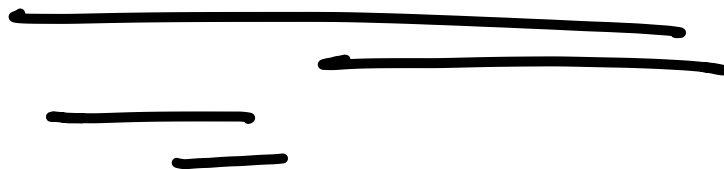
```
while(n /= 2){  
    cout<<"Halo"<<endl;  
}
```



$$n = 2^x \quad \leftrightarrow \quad x = \log_2 n$$

$\Theta(\log CN)$

$f(n/2)$



$O(2^n)$



$O(n)$

