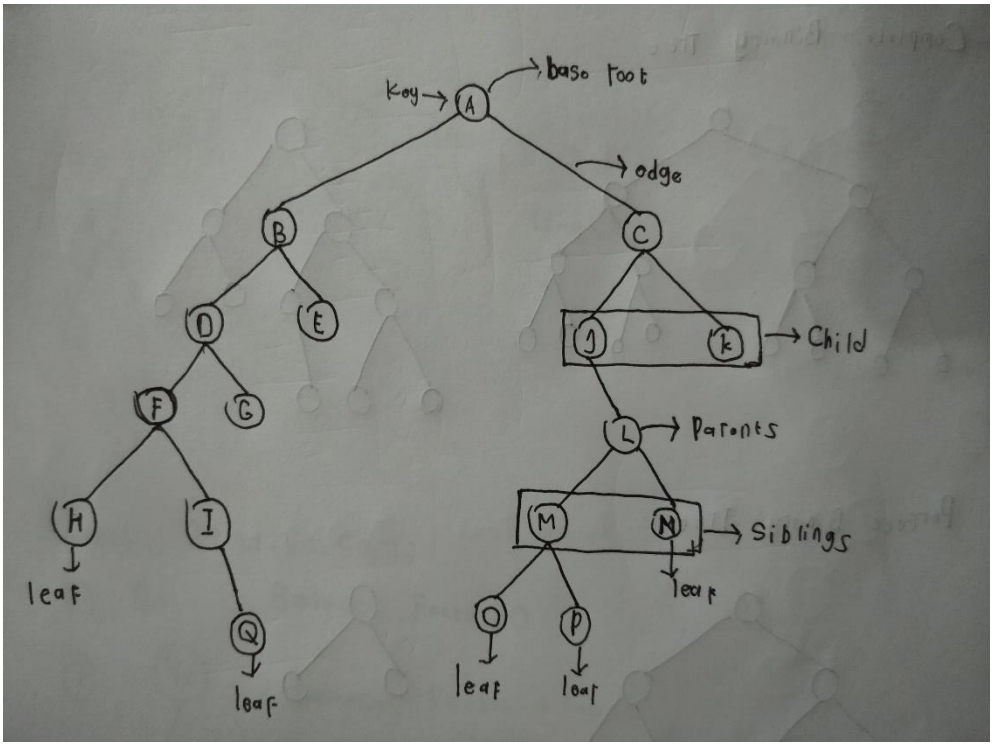
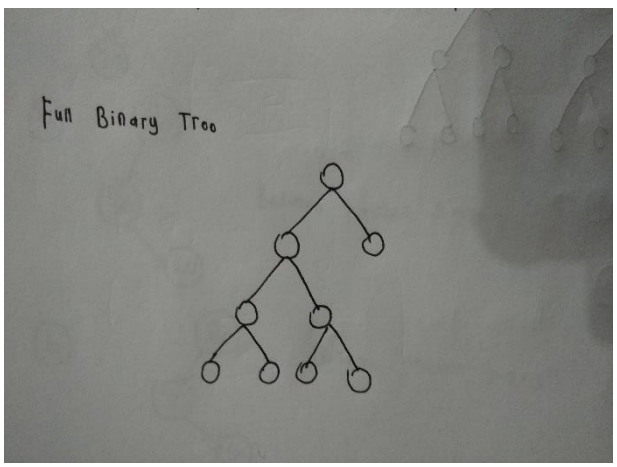
1. -Data struktur linier : Struktur data yang menyusun node secara teratur dan berdampingan, pemanfaatan memorinya tidak terlalu efisien, tapi mudah untuk diimplementasikan.

   -Data struktur non linier : Struktur data yang memiliki node yang saling berhubungan cenderung sulit dalam pengimplentasiannya

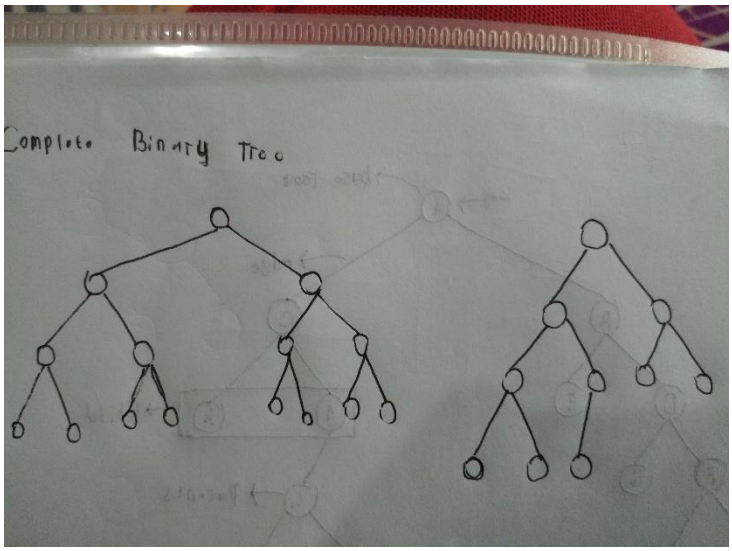
2. - Base Root: Node paling atas atau paling awal pada sebuah tree
   - Key: Nilai atau value utama dari tiap-tiap node
   - Edge: Garis pengubung antara parents dan child atau antar node.
   - Siblings: Node yang satu level dan memiliki 1 parent yang sama
   - Parent: Merupakan node yang memiliki 1 atau lebih percabangan ke node di bawahnya sebagai childnya.
   - Child: Merupakan node yang merupakan percabangan atau keturunan dari parent
   - Leaf: Node yang tidak memiliki percabangan anak di bawahnya



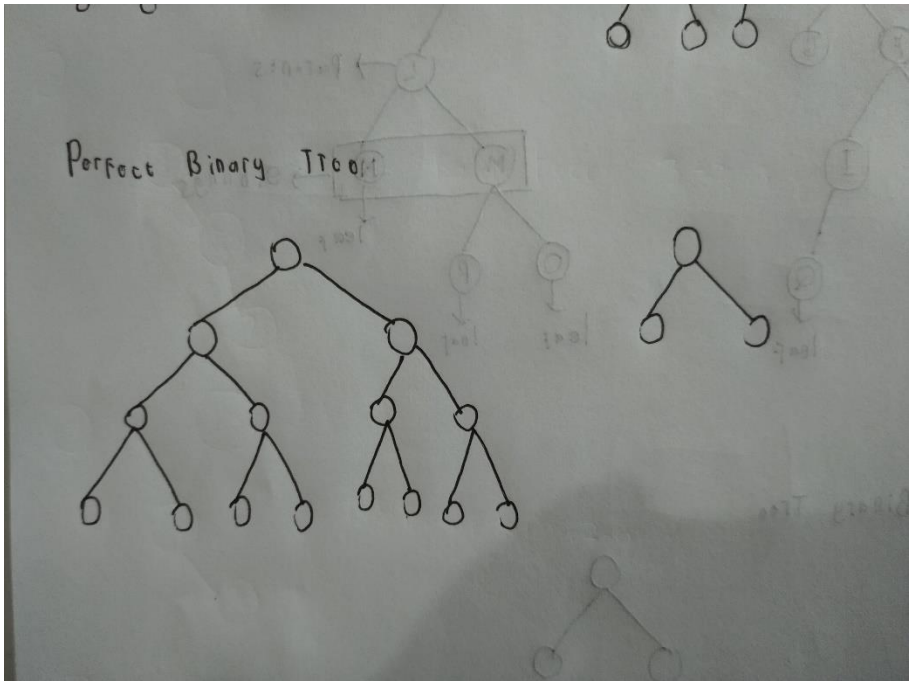
3. -Full: Binary tree yang memiliki 0 atau 2 anak



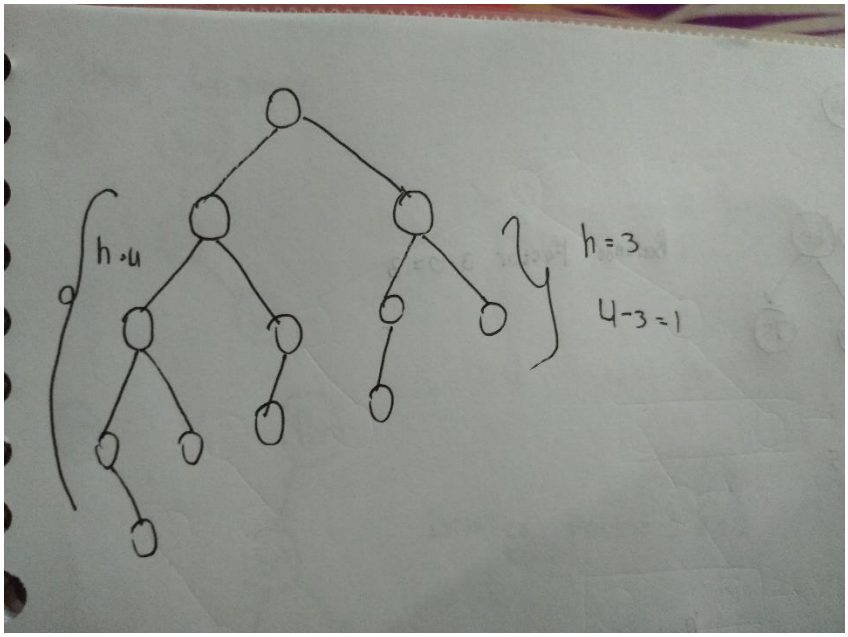
-Complete: Binary tree yang semua levelnya terisi dengan node kecuali level paling bawah.



-Perfect: Binary tree yang semua node hanya boleh punya 2 anak kecuali node paling bawah dan leaf node nya berada di level yang sama

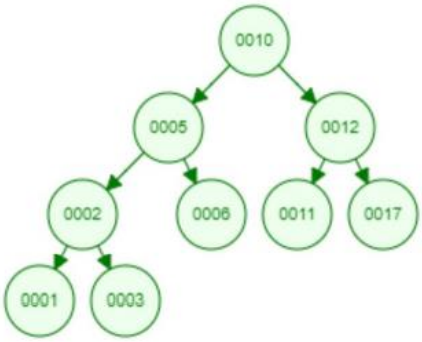


4. Jumlah node dibagi 2 balanced dan unbalanced/degenerate //degenerate: tiap node punya 1 anak kecuali node yg terakhir Balanced binary tree memiliki syarat 0(log)N dimana N adalah jumlah node atau bisa dilihat dari struktur tree itu sendiri di mana selisih tinggi substree kiri dan substree kanan maksimal 1.



5. Berlaku untuk perfect binary tree //properti jumlah node
   - Jumlah maksimum node di level k adalah 2 k . k untuk level
   - Jumlah maksimum node di binary tree/keseluruhan adalah 2 h+1 -1. h untuk height properti tinggi binary tree
   - Tinggi maksimum dari sebuah binary tree yang terdiri dari n nodes adalah n-1. n untuk banyak node
   - Tinggi minimum dari sebuah binary tree yang terdiri dari n nodes adalah 2 log(n). n untuk banyak node

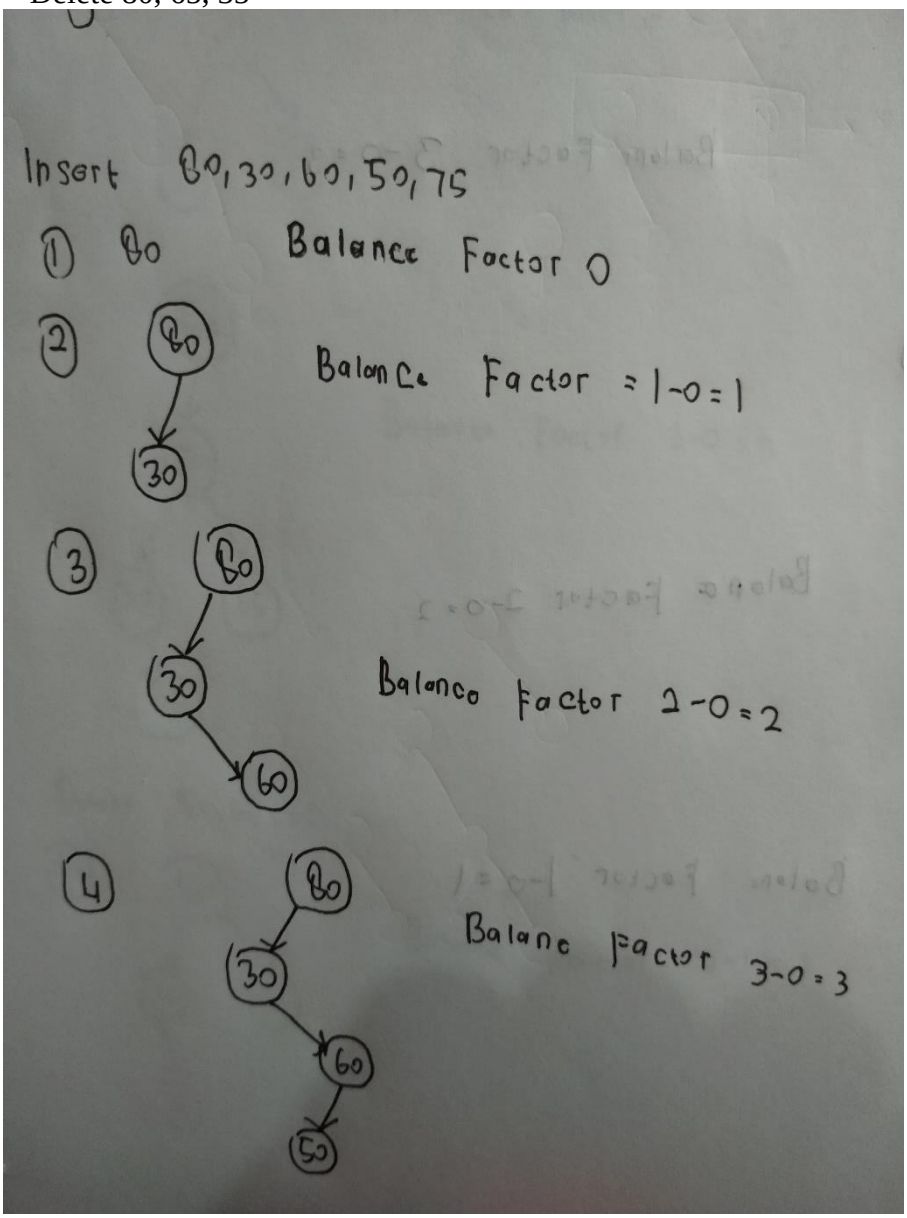
6. Array berukuran statis dan kita menggunkaan indexing dengan beberapa rumus
   - Root: index 0
   - Left child adalah 2p+1
   - Right child adalah 2p+2
   -Parent adalah(p-1)/2 // p adalah index parent



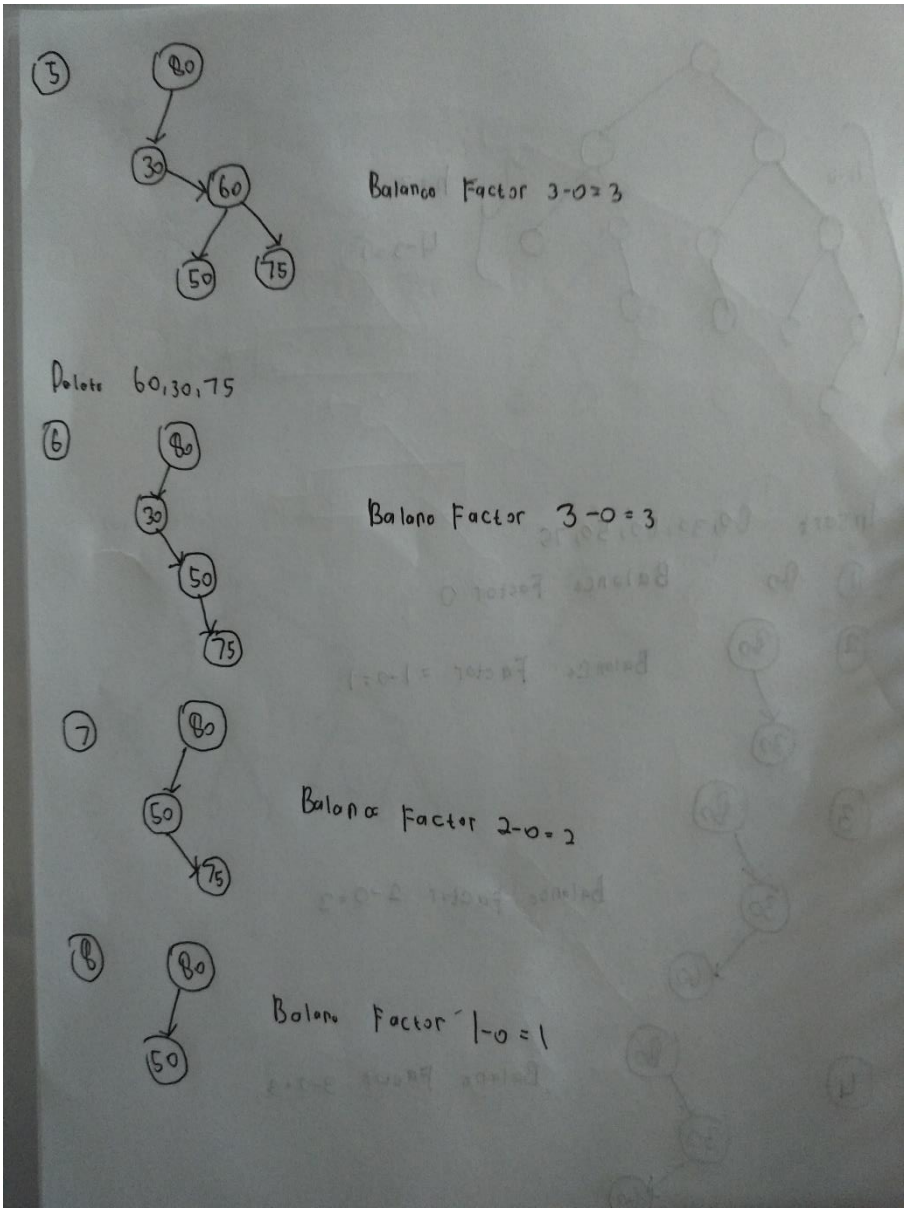
0	1	2	3	4	5	6	7	8
0010	0005	0012	0002	0006	0011	0017	0001	0003

7. Inorder successor : Node yang berada di bawah node tertentu
   Inorder predecessor : Node yang berada di atas node tertentu

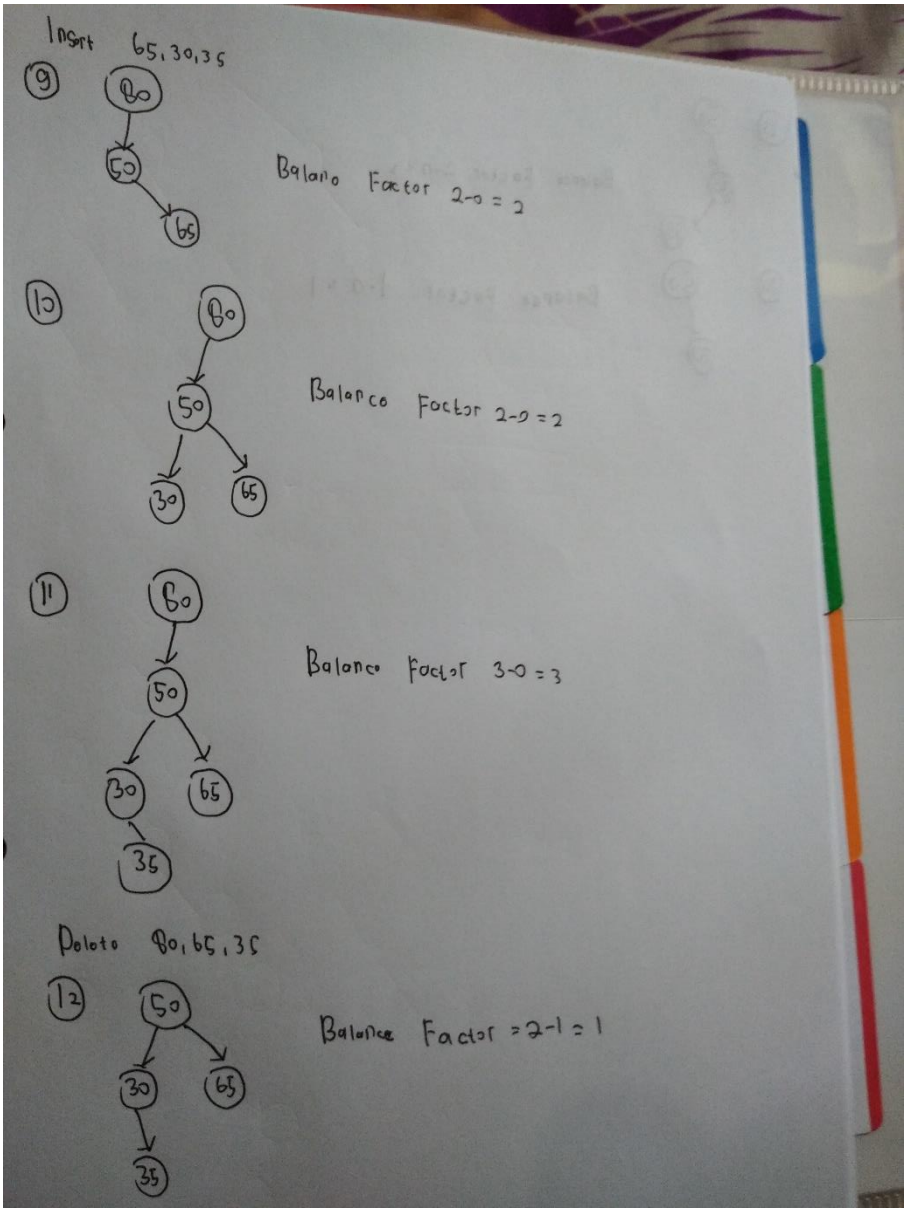
8. -Insert 80, 30, 60, 50, 75
   -Delete 60, 30, 75
   - Insert 65, 30, 30
   - Delete 80, 65, 35

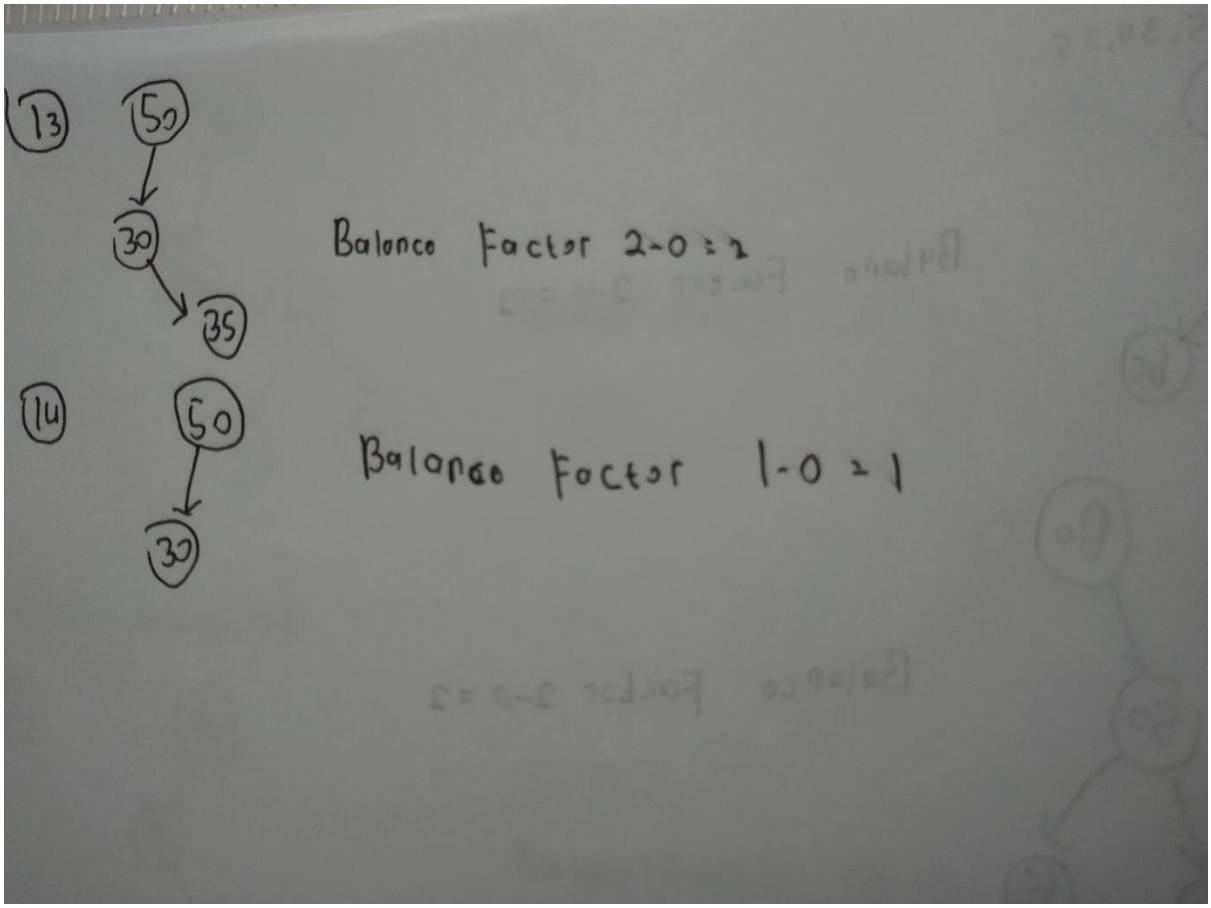


Cheryl Almeira/ ak4869#1991 / Leviathan



Cheryl Almeira/ ak4869#1991 / Leviathan




13
50
30
35
Balance Factor 2-0 = 2
14
50
30
Balance Factor 1-0 = 1