

1.

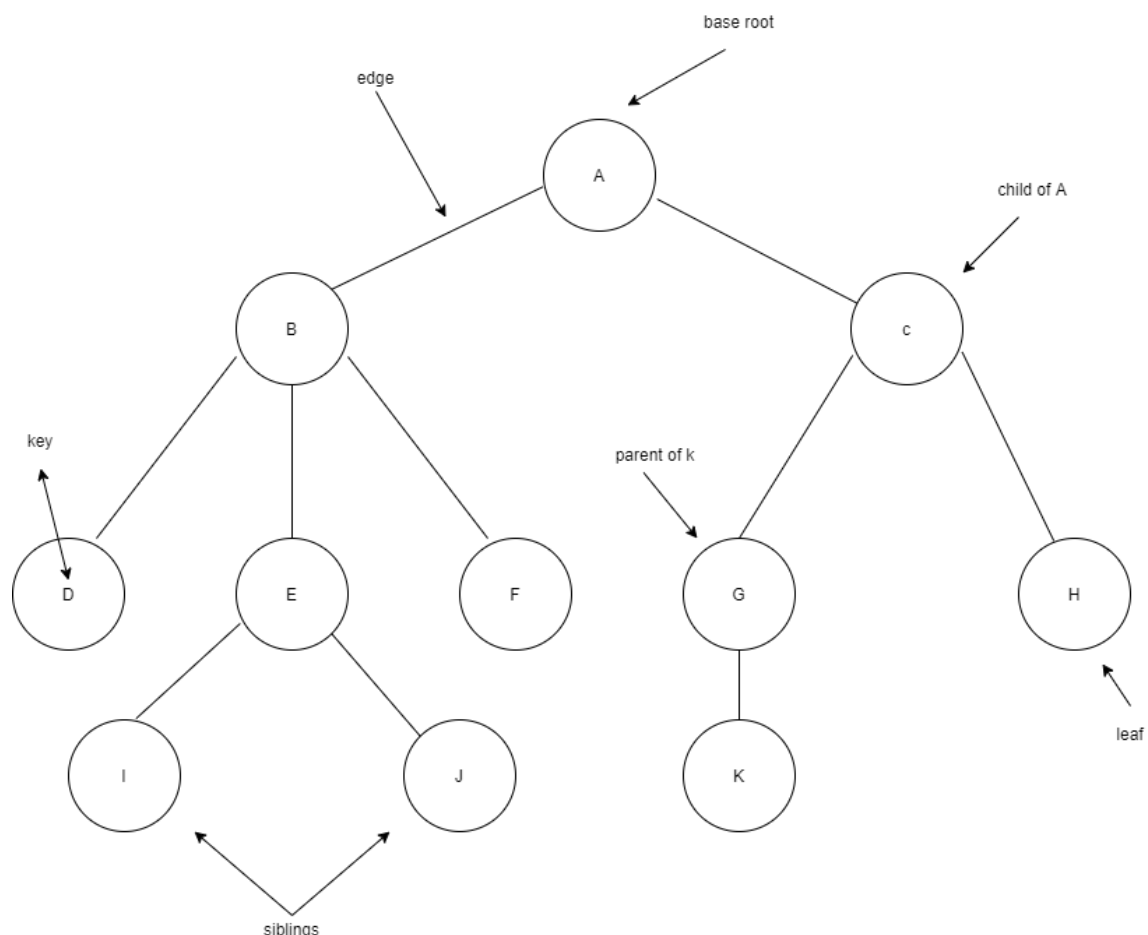
### Struktur Data Linear:

Struktur data yang elemen datanya disusun secara berurutan atau linier. Dalam struktur data linier, satu level terlibat. Oleh karena itu, kita dapat melintasi semua elemen hanya dalam satu proses. Struktur data linier mudah diimplementasikan karena memori komputer diatur secara linier. Contohnya adalah array, stack, queue, linked list, dll.

### Struktur Data Non-linear:

Struktur data yang elemen datanya tidak tersusun secara berurutan atau disebut struktur data non linier. Dalam struktur data non-linier, level tunggal tidak dilibatkan. Oleh karena itu, kami tidak dapat melintasi semua elemen dalam satu proses saja. Struktur data non-linier tidak mudah diimplementasikan dibandingkan dengan struktur data linier. Ini menggunakan memori komputer secara efisien dibandingkan dengan struktur data linier. Contohnya adalah tree dan graphs.

2.



### 3. Full binary tree

Ini adalah jenis khusus dari binary tree yang memiliki nol anak atau dua anak. Ini berarti bahwa semua node dalam binary tree itu harus memiliki dua node turunan dari node induknya atau node induk itu sendiri adalah node daun atau node eksternal. Full Binary tree adalah binary tree unik di mana setiap simpul kecuali simpul eksternal memiliki dua anak. Jika ia menampung satu anak, binary tree seperti itu tidak akan menjadi full binary tree. Di sini jumlah simpul daun sama dengan jumlah

simpul internal ditambah satu. Persamaannya seperti  $L = I + 1$ , di mana  $L$  adalah jumlah node daun, dan  $I$  adalah jumlah node internal.

### Complete binary tree

Complete binary tree adalah jenis binary tree spesifik lainnya di mana semua tree level diisi seluruhnya dengan node, kecuali tree level yang paling rendah. Juga, di level terakhir atau terendah dari binary tree ini, setiap node mungkin berada di sisi kiri.

### Perfect binary tree

Binary tree dikatakan 'perfect' jika semua simpul internal memiliki dua child, dan setiap simpul eksternal atau leaf node berada pada tingkat yang sama atau kedalaman yang sama dalam sebuah tree. Perfect binary tree yang memiliki ketinggian ' $h$ ' memiliki  $2^h - 1$  node.

4. a. The left and right subtrees' heights differ by at most one.

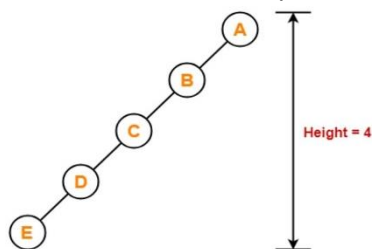
b. The left subtree is balanced.

c. The right subtree is balanced

5. a. Minimum number of nodes in a binary tree of height  $H = H + 1$

Contoh :

To construct a binary tree of height = 4, we need at least  $4 + 1 = 5$  nodes.



b. Maximum number of nodes in a binary tree of height  $H = 2^{H+1} - 1$

contoh :

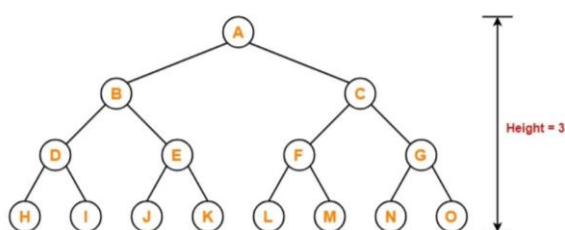
Maximum number of nodes in a binary tree of height 3

$$= 2^{3+1} - 1$$

$$= 16 - 1$$

$$= 15 \text{ nodes}$$

Thus, in a binary tree of height = 3, maximum number of nodes that can be inserted = 15.



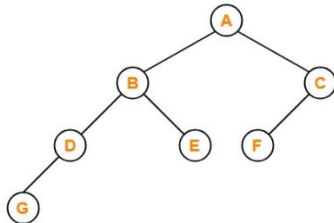
c. Total Number of leaf nodes in a Binary Tree= Total Number of nodes with 2 children + 1

contoh :

Number of leaf nodes = 3

Number of nodes with 2 children = 2

Clearly, number of leaf nodes is one greater than number of nodes with 2 children.



d. Maximum number of nodes at any level 'L' in a binary tree=  $2^L$

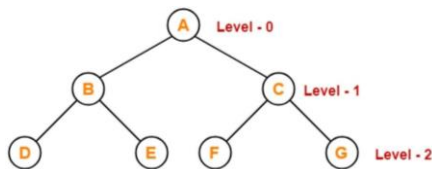
contoh :

Maximum number of nodes at level-2 in a binary tree

$$= 2^2$$

$$= 4$$

Thus, in a binary tree, maximum number of nodes that can be present at level-2 = 4.



6. a. Digunakan untuk mengecek binary tree dalam kondisi kosong atau terisi node.

b. Pengecekan menggunakan variabel size.

c. Mengembalikan true jika size = 0.

d. Mengembalikan false jika nilai size > 0.

## 7. Inorder successor

Di Binary Tree, inorder successor dari sebuah node adalah node berikutnya dalam Inorder traversal dari Binary Tree. Inorder successor adalah NULL untuk node terakhir dalam Inorder traversal.

Dalam Binary Search Tree, Inorder Successor dari sebuah node input juga dapat didefinisikan sebagai node dengan kunci terkecil lebih besar dari kunci node input. Jadi, terkadang penting untuk menemukan node berikutnya dalam urutan yang diurutkan.

## Inorder predecessor

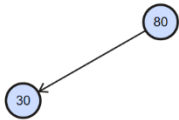
adalah node yang di-travers sebelum diberikan node dalam inorder traversal dari binary tree. Dalam pohon penelusuran biner, ini adalah nilai besar sebelumnya sebelum node. Misalnya, inorder pendahulu node (6) di bawah pohon akan 5 dan untuk node (10) adalah 6.

8.

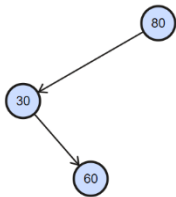
Insert 80



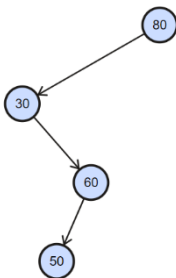
Insert 30



Insert 60

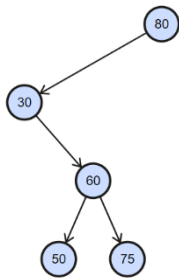


Insert 50

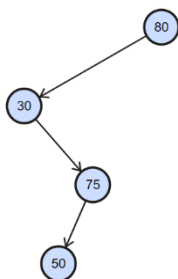


Insert 75

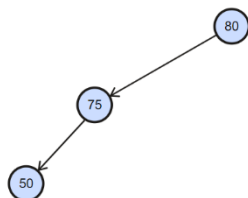
Vincent – G aja #9441



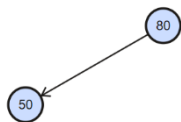
Delete 60



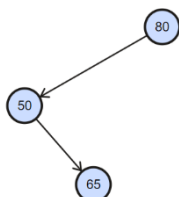
Delete 30



Delete 75

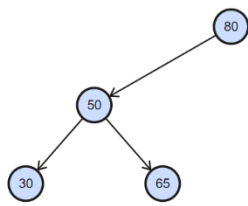


Insert 65

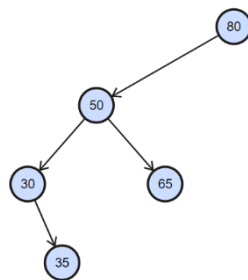


Insert 30

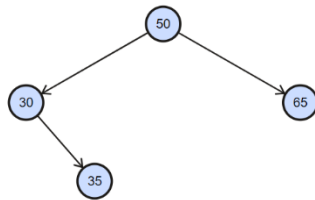
Vincent – G aja #9441



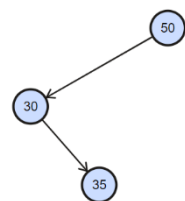
Insert 35



Delete 80



Delete 65



Delete 35

