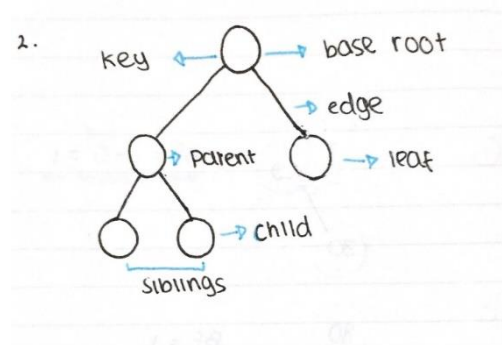


Individual Assignment

1. Explain the differences between linear and non-linear data structures!

- linear : elemen data terhubung secara berurutan dan setiap elemen dapat dilalui melalui satu proses.
- non - linear : elemen data terhubung secara hierarki dan hadir di berbagai tingkatan.

2. Describe the following terminology in a tree: base root, key, edge, siblings, parent, child, and leaf!



3. Explain the following types of binary trees: full, complete, and perfect!

- full : memiliki 0/2 anak
- complete : memiliki 0/2 semua level mesti keisi semua kec yg level terakhir
kec level terakhir tdk apa tdk diisi
dengan $bf \leq 1$
- perfect : harus 2 anak kecuali level terakhir dan semua leaf di level yang sama

4. What makes a tree balanced?

- perbedaan selisih height right dan height left setara atau selisih sama dengan 1

$$\text{Balanced Factor} = | \text{height}(\text{root} \rightarrow \text{left}) - \text{height}(\text{root} \rightarrow \text{right}) |$$

$$\text{balance : } BF \leq 1$$

$$\text{unbalance: } BF > 1$$

5. Explain the four properties of a binary tree!

- dibagi 2 yaitu number of nodes & height of tree
- number of nodes
- max nodes at level k

$$\text{rumus : } 2^k$$

- max nodes in a tree with level k

rumus : $2^{k+1} - 1$

- height of tree

- min level untuk n nodes

rumus : $\log_2(n)$

$a^b=c \rightarrow$ yg dicari b nya

- max level untuk n nodes

rumus: $n-1$

6. Explain the intuition of implementing a binary tree using an array!

- left = $2p+1$
- right = $2p+2$
- index = $2p$
- * p = parent

7. Explain the differences between inorder successor and inorder predecessor!

- Saat Anda melakukan inorder traversal dari pohon biner, tetangga dari node yang diberikan disebut Predecessor (node berada di belakang node yang diberikan)

dan Penerus (simpul terletak di depan simpul yang diberikan).

8. Draw the following binary search tree step by step (14 pictures):

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

Notes for number 8: for each picture, please write the balance factor of each node. Balance factor is defined as Balance Factor = $|\text{height of left subtree} - \text{height of right subtree}|$.

8. - insert : 80, 30, 60, 50, 75

- delete : 60, 30, 75

- insert : 65, 30, 35

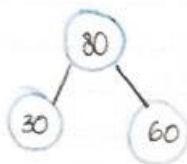
- delete : 80, 65, 35

①

80

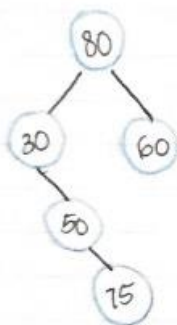
Bf = 0

③



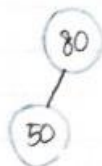
Bf = 0

⑤



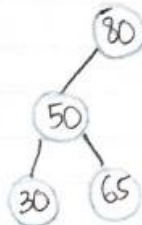
Bf = 2

⑦



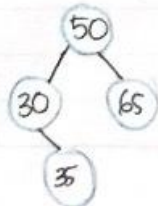
Bf = 1

⑩



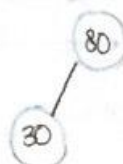
Bf = 2

⑫



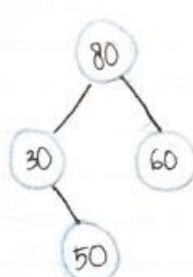
Bf = 2

②



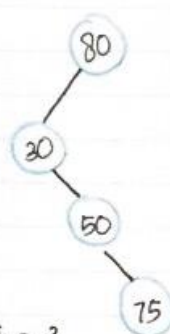
Bf = 1 - 0 = 1

④



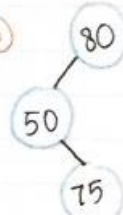
Bf = 1

⑥



Bf = 3

⑧



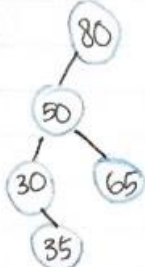
Bf = 2

⑨



Bf = 2

⑪



Bf = 3

⑬



Bf = 2

⑭



Bf = 1