Username Discord	**: Vincent Jonathan – 2401962024#2211**
Grup Bootcamp	**: G9J**
Tentang	**: Individual Assignment**
Tanggal	**: 19 Januari 2021**

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

- Linear data structure is type of data structure which is the node interact one each other in one row (linear). The example is such as single/double linked list, stack, queue, and skewed/unbalanced/degenerated binary tree. This type of data struct will become one row if we visualize them as tree (height/deep of tree = node amount – 1, for tree level start from 0).

- Non-linear data structure is type of data structure which is the node could interact with more than one node directly. The example such as Tree which have branches and have leaf more than 1. This type of data structures won't become one row if we visualize them as tree. (height/deep of tree < node amount – 1, for tree level start from 0).

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

- Base Root is the ancestors of all root (The source of the tree)
- Key is the name or value of a node/vertex/root
- Edge is the name of line which is connect parent and child
- Siblings is the name of the keys which is in the same level of tree with the same parent
- Parent is the direct root of the key
- Child is the direct key which have 1 deeper level.
- Leaf is the key which don't have child.

Example:

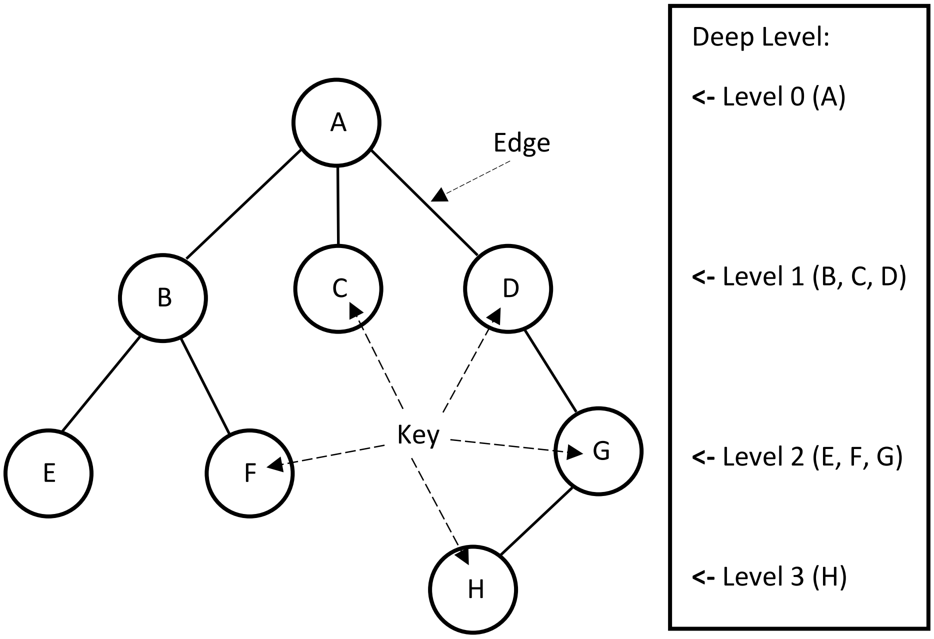
- Base Root = {A}

- Sibling of B = {C, D}
  Sibling of F = {E}
  Sibling of G = {}

- Parent of B, C, D is A
  Parent of E, F is B
  Parent of G is D
  Parent of H is G

- Child of A = {B, C, D}
  Child of B = {E, F}
  Child of D = {G}
  Child of G = {H}

- Leaf = {C, E, F, H}



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

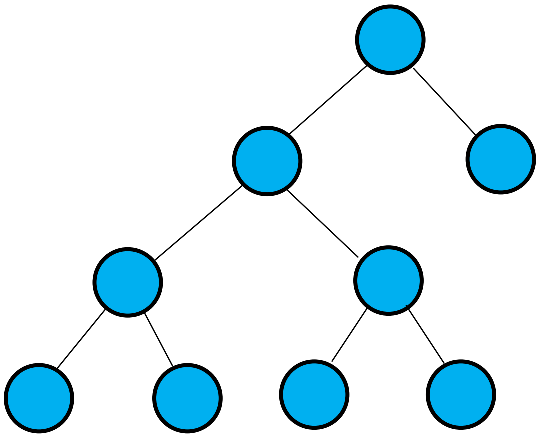
- Full is type of binary tree which there's no any key that have 1 child (only 0 or 2 child)

- Complete is type of binary tree which all level of the tree must have 2 child except the deepest level (must have 0 child) and the deepest level – 1 (can't have 2 child). Beside of it, the child of the deepest level – 1 must fulfill from the left hand of the tree (before the left side in a level have child, the right side can't have child).

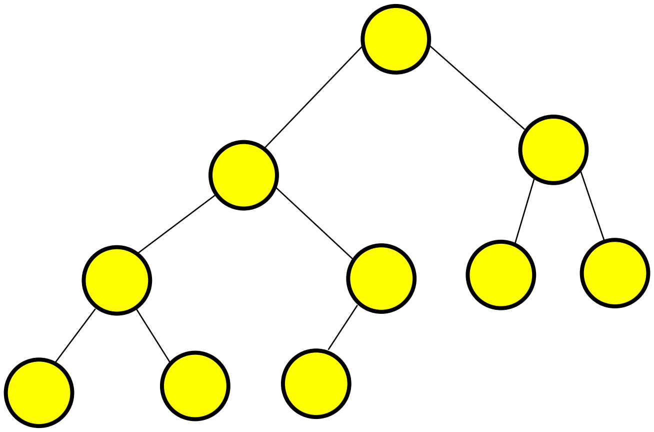
- Perfect is type of binary tree which have node/vertex amount equal with $2^{k+1} - 1$ (k is level of tree, start from 0)

As example:

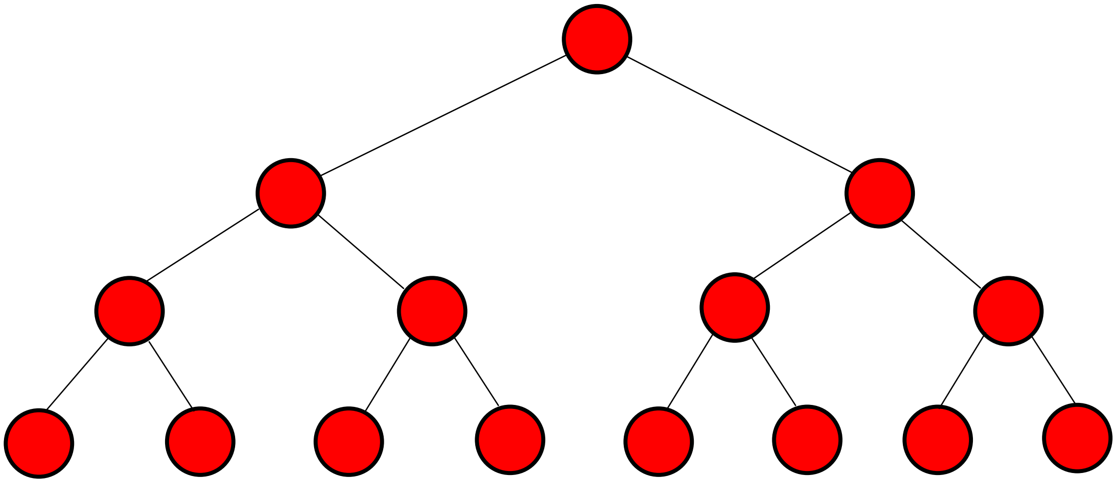
- Full Binary Tree



- Complete Binary Tree



- Perfect Binary Tree



4. What makes a tree balanced?

- Tree level amount must ${}^{10}\log_2$ x n (n = amount of node/vertex)
- Each left and right descendant amount of node/vertex must have difference at max 1 (Only count till 2 level deeper or 2 generation)

5. Explain the four properties of a binary tree!

- Each deep level of tree has their own maximum node/vertex (Maximum vertex formula: (2^{k}), k is deep level which start from 0)
- Maximum node/vertex amount in a binary tree is 2^{k+1} – 1, with k is the level of tree (start from 0)
- Minimum level of tree is ${}^{2}log_{n}$ with n is the amount of nodes/vertex in the tree
- Maximum level of tree is n-1 with n is the amount of nodes/vertex in the tree

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

- Binary tree could be implemented using array by use the index. Base root of binary tree should be filled in the first of array (on index 0). Left child of a root could be found in index 2.P+1 and the right child could be found in index 2.P+2 with P is the root index. Parent of a child could be found in index (P-1)/2 with P is the child index.

7. Explain the differences between in-order successor and in-order predecessor!

- In-order successor is type of deletion in binary tree which the deleted key/value will changed by the lowest value in a tree, but higher than the deleted value and find out by using in-order technique

- In-order predecessor is type of deletion in binary tree which the deleted key/value will changed by the highest value in a tree, but lower than the deleted value and find out by using in-order technique.

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 = |height of left subtree - height of right subtree|.

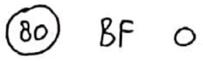
Tanggal: 19 Januari 2021
Tentang : Individual Assignment
Mata Kuliah: Bootcamp - Data Structures
Nama : Vincent Jonathan

Keterangan:

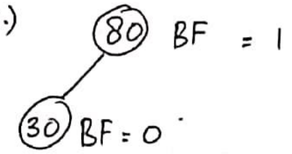
- Balance Factor dilambangkan "BF"

8.) Insert 80, 30, 60, 50, 75

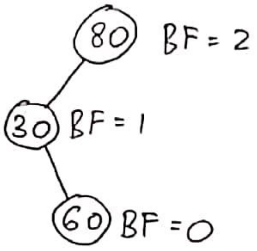
1.)



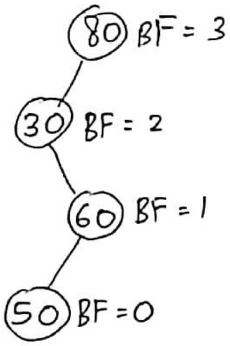
2.)



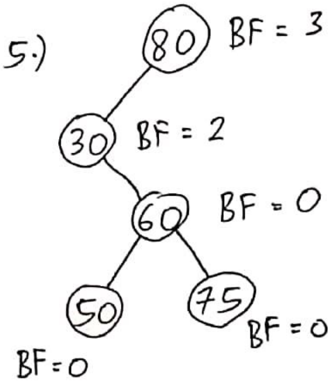
3.)



4.)

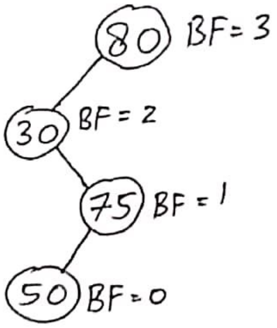


5.)

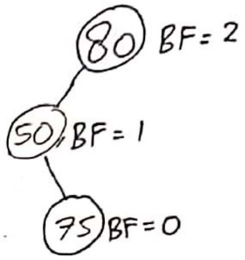


Delete 60,30,75

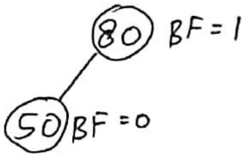
6.)



7.)

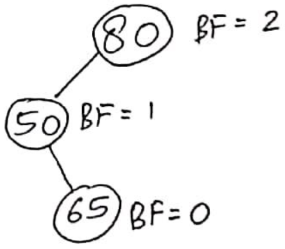


8.)

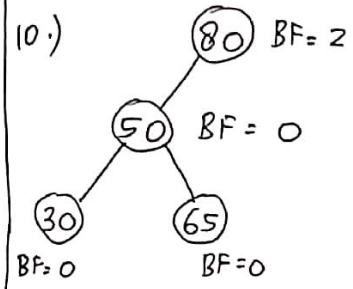


Insert 65, 30, 35

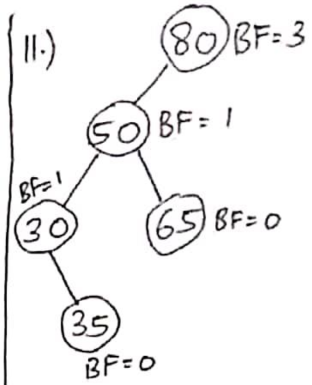
9.)



10.)

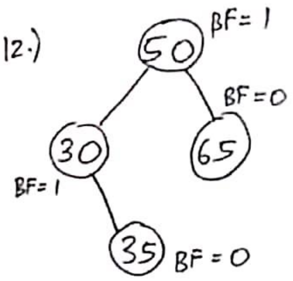


11.)

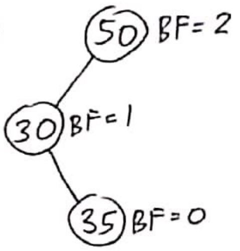


Delete 80, 65, 35

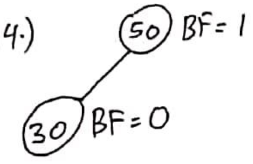
12.)



13.)



14.)


50 BF = 1
30 BF = 0