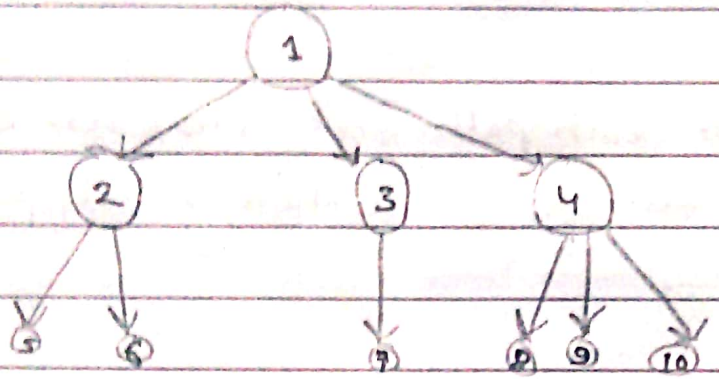
Date / /
Page no

Tree Data structure

- A tree is a non linear hierarchical data structure that consists of nodes connected by edges.



Why Tree Data structures?

Other DS such as arrays, stack, queue are linear datastructure that stores data sequentially. In order to perform any operation in a linear datastructure, the time complexity increases with increase in the data size. But it is not acceptable in today's computational world.

Different datastructure allow quicker and easier access to the data as it is a non-linear data structure.

Tree terminologies:

① Node

A node is an entity that contains a key or value & pointer to its child nodes.

The last nodes of each path are called leaf nodes or external nodes that do not contain a link/pointer to its child.

The node having at least a child node is called an internal node.

② Edge

→ It is the link between two nodes.

③ Root

It is the topmost node of a tree.

④ Height of a node:

The height of a node is the number of edges from the ~~root~~ node to the deepest leave (ie longest path from the node to a leaf node).

⑤ Depth of a Node:

→ Number of edges from root to that node.

⑥ Height of a Tree:

→ The height of the Tree is the height of the root node or depth of the deepest node.

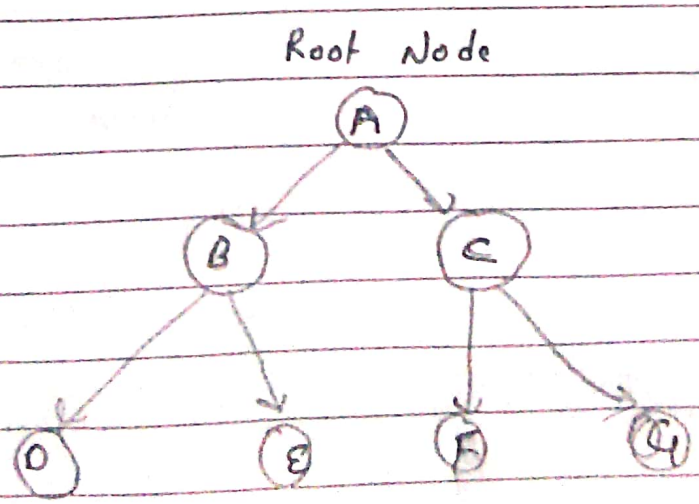
⑦ Degree of a Node:

→ Total number of branches of that node.

• Types of Tree in Datastructure:

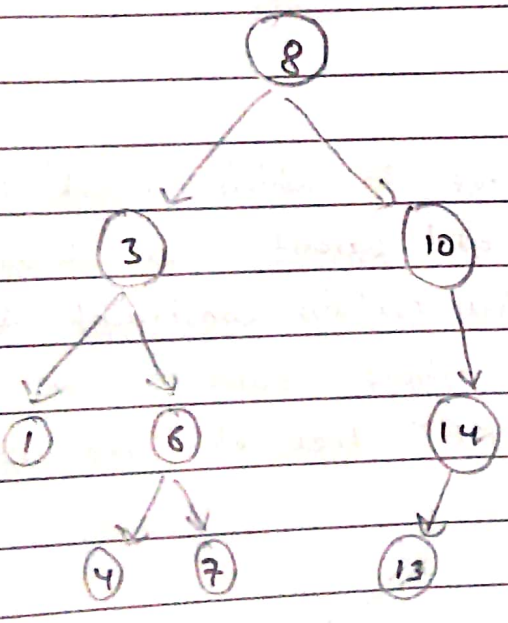
① Binary tree:

It is the kind of tree in which most two children can be found for each parent. The kids are known as left kid & right kid. When certain constraint & characteri are applied, a number of others such as AVL Tree, BST (Binary Search tree), RBT tree etc are also used.



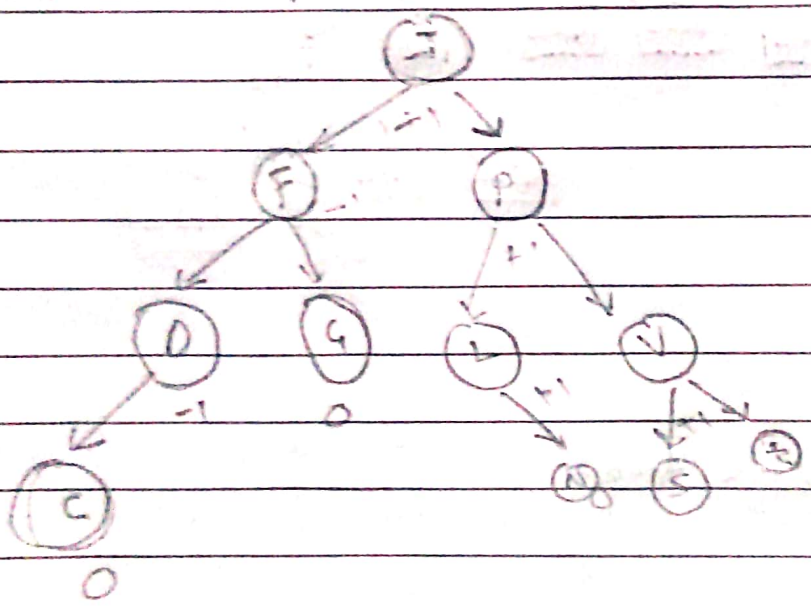
(2) Binary Search Tree (BST)

→ BST is a binary tree extension with several optional restrictions. The left child value of the node should in BST be less than or equal to parent value & the right child value should always be greater than or equal to parent's value. The BST property makes it ideal for search operations since we can accurately determine at each node. whether the value is in left or right sub-tree. That's why search tree is named.



③ AVL Tree

AVL Tree is a Binary search tree self balancing on behalf of the inventors Adelson-Velshi & Landis, the name AVL is given. This was first tree that balanced dynamically. A balancing factor is allocated for each node in the AVL Tree, based on whether the tree is balanced or not. In AVL Tree the correct balance factor is 1, 0, & -1.



④ Red-Black Tree

Another kind of auto-balancing tree is red black According to red-black tree's properties, the red black name is given because the red black tree has either red or black painted in each node. It maintains the balance of forgt. Even though this tree is not fully balanced, the searching operation only takes O(log n) time. When the new nodes are added in Red-black Tree, nodes will be rotated to maintain the red-Black Tree's properties.

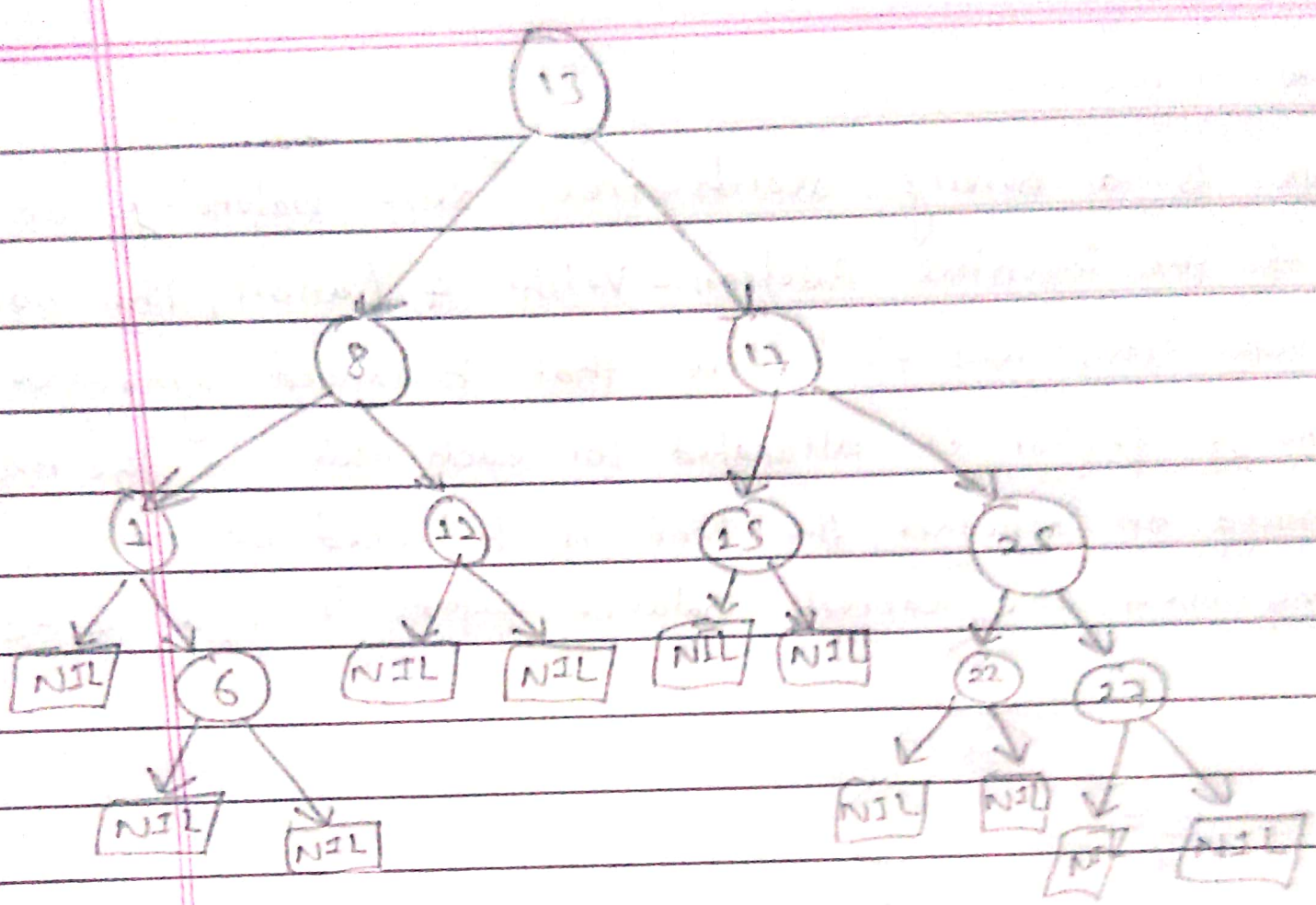


fig: Red black tree.