Data Structure Theory Assignment:- Tree

Name:- Sumit Sunil Koundanya PRN:-2019BTEIT00023

Q.1) what is Tree and explain Tree Terminology with example

A tree is a hierarchical data structure defined as a collection of nodes. Nodes represent value and nodes are connected by edges. A tree structure means that the data are organized so that items of information are related by branches.

A tree is a finite set of one or more nodes such that There is a specially designated node called root. The remaining nodes are partitioned into n>=0 disjoint set T1,…,Tn, where each of these sets is a tree. T1,…,Tn are called the sub trees of the root.

* Every node in the tree is the root of some sub- tree

**Terminologies related to Tree:-**

* + **Node**: the item of information plus the branches to each node.
  + **Degree**: the number of subtrees of a node
  + **Degree of a tree**: the maximum of the degree of the nodes in the tree.
  + **Terminal Nodes (or leaf):** nodes that have degree zero
  + **Non- terminal nodes**: nodes that don’t belong to terminal nodes.
  + **Children**: the roots of the subtrees of a node X are the children of X
  + **Parent**: X is the parent of its children.
  + **Siblings**:-Two or more nodes with same parent
  + **Ancestors of a node**: all the nodes along the path from the root to that node.
  + **The level of a node**: defined by letting the root be at level one. If a node is at level l, then it children are at level l+1.
  + **Height (or depth):** the maximum level of any node in the tree

Example:-

A is the root node The level of E is 3

B is the parent of D and E The height (depth) of the tree is 4

C is the sibling of B The degree of node B is 2

D and E are the children of B The degree of the tree is 3

A, B, C, H are internal nodes The ancestors of node I is A, C, H

D, E, F, G, I are external nodes, or leaves The descendants of node C is F, G, H, I

Q.2) What is Binary Tree and explain different Tree Traversal Methods.

A Binary Tree is a tree in which No node can have more than 2 sub-tree. A binary tree is a finite set of nodes that is either empty or consists of a root and two disjoint binary trees called the left sub-tree and the right sub-tree.

**Binary Tree Traversals:-**

1) Depth First Traversal

In Depth First Traversal, the processing proceeds along the path from root through one child to most distant descendants of that child before processing a second child. In the other world in the depth first traversal all of descendants of child are processed before next child. To traverse a child in DFT we use stack data structure. There are three types of traversal:-

A) Pre order Traversal: - In this traversal root node is processed first followed by left sub-tree and then right sub-tree. The prefix pre means to go before thus root goes before sub-tree. [Root-Left-Right]

B) In order Traversal: - In this Traversal, processes the left sub-tree first then root and finally right sub-tree. Meaning of prefix In is that root is processed in between sub-trees. [Left-Root-Right]

C) Post order Traversal: - In this traversal it processes root node after left and right sub-tree. It starts by locating left most leaf and processing it. It then processes its right sibling including its sub-tree if any, finally it processes root node. [Left-Right-Root]

2) Breadth First Traversal

In Breadth First Traversal processing processed horizontally from root to all its children, and then to its children’s children and so forth until all nodes have been proceeds. In other word, in BFT each level is completely processed before next level is started. To traverse a child in BFT we use queue data structure.

Q.3) Write an algorithm for binary tree traversal pre order, in order and post order.

A) Pre order Traversal

Algorithm Preorder(tree)

1. Visit the root.

2. Traverse the left subtree, i.e., call Preorder(left-subtree)

3. Traverse the right subtree, i.e., call Preorder(right-subtree)

B) In order Traversal

Algorithm Inorder(tree)

1. Traverse the left subtree, i.e., call Inorder(left-subtree)

2. Visit the root.

3. Traverse the right subtree, i.e., call Inorder(right-subtree)

C) Post order Traversal

Algorithm Postorder(tree)

1. Traverse the left subtree, i.e., call Postorder(left-subtree)

2. Traverse the right subtree, i.e., call Postorder(right-subtree)

3. Visit the root.

Q.4) what are the properties of Binary Tree?

**1) The maximum number of nodes at level ‘l’ of a binary tree is 2l.**

Here level is the number of nodes on the path from the root to the node (including root and node). Level of the root is 0.

**2) The Maximum number of nodes in a binary tree of height ‘h’ is 2h – 1.**   
 Here the height of a tree is the maximum number of nodes on the root to leaf path. Height of a tree with a single node is considered as 1.

**3) In a Binary Tree with N nodes, minimum possible height or**the **minimum number of levels is Log2(N+1) ?**

**4) In Binary tree where every node has 0 or 2 children, the** **number of leaf nodes is always one more than nodes with two children**.

L = T + 1

Where L = Number of leaf nodes

T = Number of internal nodes with two children

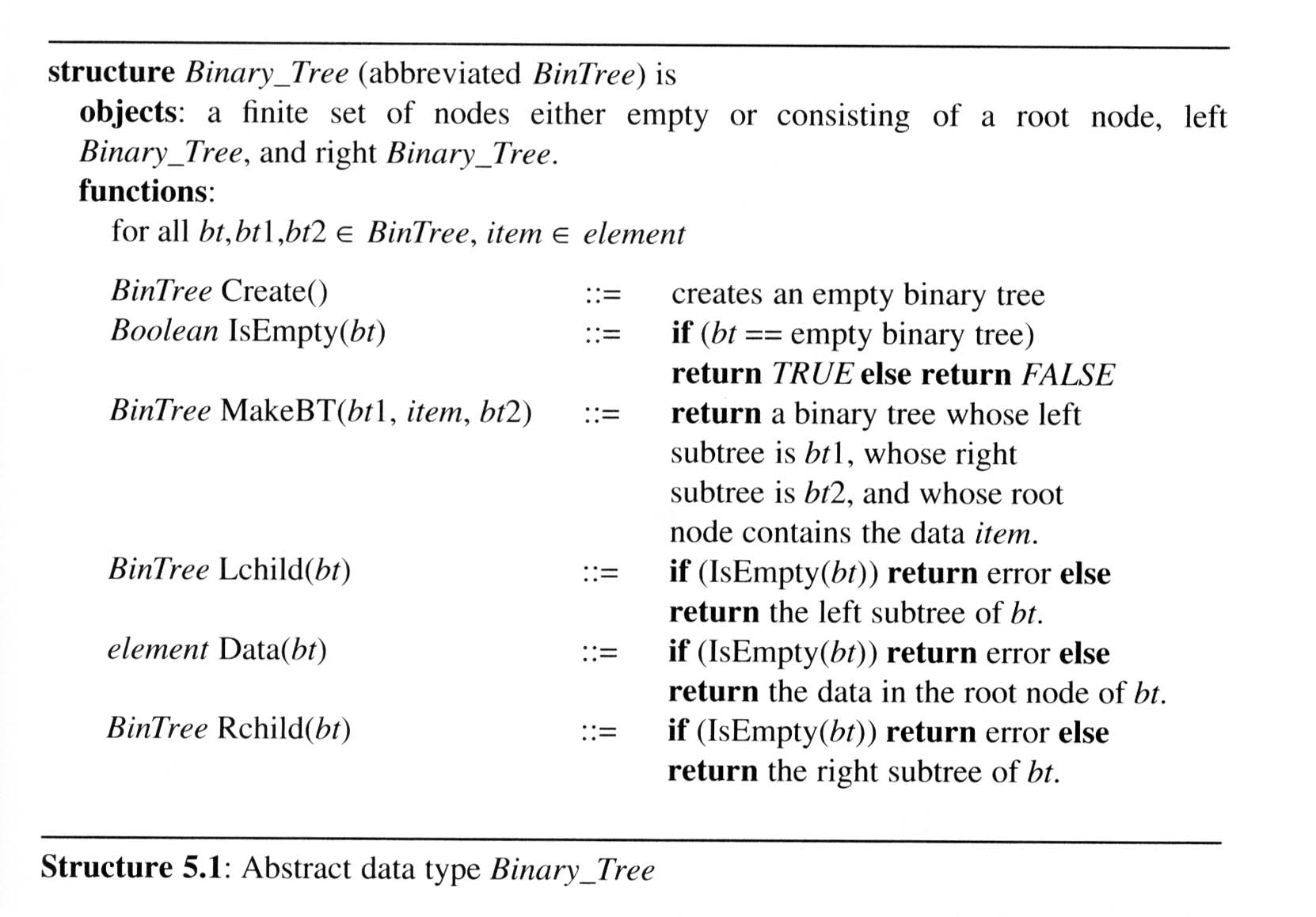
5) A Binary Tree with L leaves has at least Log2L + 1   levels   
 A Binary tree has the maximum number of leaves (and a minimum number of levels) when all levels are fully filled. Let all leaves be at level l, then below is true for the number of leaves L.

L <= 2l-1 [From Point 1]

l = Log2L + 1

where l is the minimum number of levels.

Q.5) Give the ADT of binary Tree



Binary Tree node declaration

typedef struct tree\_node \*tree\_ptr;

struct tree\_node

{

   element\_type element1;

   tree\_ptr left1; tree\_ptr right1;

};

typedef tree\_ptr TREE;

Right Sub-tree

Left sub-tree