CSP2348/CSP5243 Data Structures

Tutorial 06: Binary Tree Data Structures

Related Objectives from Unit Outline:

- Describe (arrays, linked lists, binary trees, and hash tables) data structures and analyse the complexity and performance of their associated algorithms.

Objectives:

1. To become familiar with the general properties of binary trees and binary search trees (BST), and their specific properties in Java;
2. To demonstrate the awareness of the principles of algorithms in BST insertion, deletion, searching, merging, and sorting.

Tasks:

Complete the following.

Task 1: Explain the relationship between the depth and node of a binary tree using the examples given below:

a. How many nodes does a fully-balanced binary tree of depth 4 have?
b. What is the maximum depth of an balanced binary tree of 30 nodes?
c. Verify your answers above by drawing illustrative binary trees.

Task 2: Consider a binary search tree (BST) whose elements are abbreviated names of chemical elements.

a. Starting with an empty BST, show the effect of successively inserting the following elements: H, C, N, O, Al, Si, Fe, Na, P, S, Ni, Ca.
b. Show the effect of successively deleting Si, N, O from the resulting BST.

Task 3: Test the Java implementation of a `Binary Search Tree` given in TreeTest.java (Download the Java code from Blackboard).

a. Execute this program;
b. Use the first line of values to draw this BST;
c. Hand-test the visitation of this BST in terms of Pre-order, In-order, and Post-order traversals;
d. Compare your results with the executed results.

Task 4 *(Optional):*

Given the following traversal orders, draw the binary tree.

Pre-order: A, B, D, C, E, F, G
In-order: B, D, A, E, C, G, F