Trees

Updated: 20th July, 2023

Aims

- To implement a binary tree.
- To traverse a tree.

Before the Practical

- Read this practical sheet fully before starting.
- Ensure you have completed either Practical 3 or Practical 4.

Activities

1. Binary Search Tree Implementation

Following the lecture slides as a guide, implement a Binary Search Tree using a `DSATreeNode` and `DSABinarySearchTree` class.

Note:

- `DSATreeNode` has already been written for you, but you'll need to understand and test it.
- The code for `find()` was already implemented for you - `insert()` and `delete()` are very similar. *The methods must all use the **recursive** approaches and pseudocode from the lecture slides.*
- You may want to leave `delete()` until you finish the rest of the practical and then come back to it.

2. Implement Additional Methods

The lecture slides described the approach for doing `min()`, `max()` and `height()`. Implement each of these operations in `DSABinarySearchTree`.

Now consider how you would give a percentage score for how balanced the tree is. Implement this approach as a new method called `balance()`.

> **Note:** Approaches can include comparing left and right heights or comparing potential and actual leaf nodes.

3. Implement Traversal Methods

The lecture slides described the approach for doing `inorder()`, `preorder()` and `postorder()` traversals of a tree. Add recursive implementations of these algorithms inside `DSABinarySearchTree` to output the traversed tree.

> **Note:** You may want to export the output of each traversal method as a queue or linked list, which can then be iterated over to display the contents.

4. Interactive Menu for DSABinarySearchTree

Setup an interactive menu system to explore building a binary tree from scratch. Include at least the following options:

(a) Add node

(b) Delete node

(c) Display the tree - ask the user if they want inorder, preorder or postorder traversal

Submission Deliverable

- Your code are due 2 weeks from your current tutorial session.
  - You will demonstrate your work to your tutors during that session
  - If you have completed the practical earlier, you can demonstrate your work during the next session
- You must **submit** your code and any test data that you have been using **electronically via Blackboard** under the *Assessments* section before your demonstration.
  - Java students, please do not submit the *.class files

Marking Guide

Your submission will be marked as follows:

- [6] Your DSABinarySearchTree and DSATreeNode are implemented correctly.
- [4] You have implemented and tested your methods for min(), max(), height() and balance().
- [6] You have implemented and can demonstrate your methods for inorder(), preorder() and postorder().
- [4] You have an interactive menu for the tree operations.

End of Worksheet