Faculty of Engineering, University of Jaffna,

Department of Computer Engineering

EC4070: Data Structures and algorithms

Lab 06

Chapter 05: Trees

Duration: 3 Hours Lecturer: Ms.Sujanthika M.

Instructions

i. Submit the code files and screenshot of the outputs in a zipped folder by naming as 2022EAAA_Lab06(AAA – Your Registration Number)
ii. Submit your zip file before the given deadline.
iii. Any plagiarized work will be given 0 marks.

A text editor uses a **BST** to store a dictionary of words. Each node of the BST contains a word, and the tree is organized such that words are stored in lexicographical (dictionary) order.

Tasks:

1. **Initial Setup:**
    a. Insert the following words into a BST:
    b. ["apple", "banana", "grape", "mango", "peach", "pear", "pineapple", "melon", "plum", "orange"]
    c. Display the in-order traversal of the BST to verify the words are sorted lexicographically.
2. **Basic Autocomplete Functionality:**
    a. Implement a function to retrieve all words that start with a user-specified prefix.
    b. Test the function with the following prefixes:
        i. "pe"
        ii. "pi"
        iii. "me"
3. **Dynamic Updates:**
    a. Add the following new words to the BST:
    ["peanut", "mandarin", "pistachio"]

b. Display the updated in-order traversal of the BST.
c. Retrieve suggestions for the prefix **"pe"** again after the update.

4. **Advanced Prefix Matching:**
    a. Modify the autocomplete function to handle case-insensitive searches. For example, the prefix **"PI"** should return results for **"pi"**.
    b. Test this functionality with the following prefixes:
        i. **"PI"**
        ii. **"Ap"**
        iii. **"Me"**
5. **Word Deletion:**
    a. Delete the following words from the BST:
        "pear", **"banana"**, and **"melon"**.
    b. Display the updated in-order traversal of the BST.
    c. Retrieve suggestions for the prefix **"pe"** and verify the results after deletion.