



University of Colombo School of Computing
SCS 1308 - Foundations of Algorithms
Take-Home 12

Instructions

- Try the following questions and upload your answer script as a zip file to the given link in the UGVLE on/before 23rd February at 6 pm.
 - Note: Rename your zip file with your index number and name. (i.e: indexNo_Name.zip).
-

Conceptual Questions

1. Explain the key properties of a Binary Search Tree (BST).
 2. Describe the process of deleting a node with two children in a BST. Illustrate with an example.
 3. Provide recursive formulas for computing the height and size of a BST.
 4. Write pseudocode to find the minimum and maximum values in a BST.
-

Programming Tasks

Task 1: BST Implementation

Write a program to implement a BST with the following functions:

- Insert a node
- Find the minimum and maximum node
- Compute the size of the tree
- Compute the height of the tree

Task 2: Deletion in BST

Extend your BST program to support node deletion. Ensure that all three deletion cases are handled:

- Deleting a leaf node
- Deleting a node with one child
- Deleting a node with two children

- **Task 3: Mirror Image of BST**

Write a function to generate the mirror image of a given BST and display it.

Task 4: DSW Algorithm for BST Balancing

Implement the Day-Stout-Warren (DSW) algorithm to balance an unbalanced BST. Your implementation should:

- Convert a BST into a right-skewed tree (vine)
 - Convert the vine into a balanced BST
-

Analysis Questions

1. Compare the performance of searching in a BST vs. a linked list. When does a BST degenerate into a linked list?
 2. Discuss the differences between the DSW algorithm and AVL trees in maintaining balance. Provide an example where DSW is preferable.
 3. Analyze the time complexity of the DSW algorithm and justify why it is an efficient balancing method.
-