PDS Tutorial Questions

Tutorial/Prep 9

Sep. 23, 2024

Information

- This document comprises tutorial questions for CS2700, which includes both conceptual/theory questions (relevant for CS2700) and programming questions (relevant for both CS2700 and CS2710; these questions can also be thought of as preparatory/practice programming questions for CS2710 Lab 9, and so also referred to as Prep 9).

Conceptual questions (for CS2700)

1. **Insertions and Deletions in a BST:** You are given an empty Binary Search Tree (BST). Insert the following elements in the given order:

   $$50, 30, 70, 20, 40, 60, 80$$

   After completing the insertions, perform these deletions in order: Delete(20), Delete(30), and Delete(50).

   Draw the structure of the BST after each insertion and each deletion.

2. **Binary Tree Construction from Traversals:** You are given the following pre-order and in-order traversals of a binary tree:

   - **Pre-order traversal:** 40, 20, 10, 30, 60, 50, 70
   - **In-order traversal:** 10, 20, 30, 40, 50, 60, 70

   Reconstruct the binary tree from these traversals and draw its structure.

3. **Kth Smallest Element in a BST**: Given a Binary Search Tree (BST), write a function in pseudocode to find the **k-th smallest element** in the BST for a given k.

4. **Height and Performance of a BST**: Consider a BST constructed over n nodes.

   1. **Height Analysis:** What is the maximum height of a BST with n nodes? Describe the specific case of insertions that leads to this height.
   2. **Time Complexity:** In the worst-case scenario, what are the time complexities for search, insertion, and deletion operations in this configuration?
   3. **Improvement Strategies:** Discuss potential strategies or alternative data structures that could be employed to improve the performance of the BST, particularly in maintaining a more balanced structure. (**Hint:** Consider techniques that can be applied after each operation to adjust the tree structure and ensure that the height remains manageable.)

5. Prove that in a linked data structure implementing a binary tree with N nodes, there are N+1 null pointers (i.e., N+1 of the children of these N nodes hold nullptr)".

Programming questions (for CS2710 Lab 9 preparation/practice, and for CS2700)

6. [IDENTICAL OR NOT] Given two binary trees you have to tell whether they are identical or not. Two trees are identical when they have the same data and the arrangement of data is also the same.

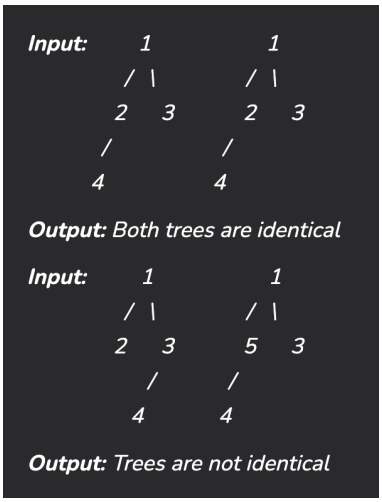

Figure 1: example

7. [TREE WARMUP] Write a program to parse a binary tree from Newick format, and a function to compute the height of this binary tree.
8. [RECONSTRUCT] Reconstruct a binary tree from its inorder and preorder traversal. See question Q2 above.
9. [ROOT TO LEAF PATH] Given a leaf in a general (rooted) tree, write a function to print the path from the root to this leaf.
10. [INVERT A BINARY TREE] Write code to create a mirror image of (invert) a binary tree, such that the left and right children of all nodes are interchanged.

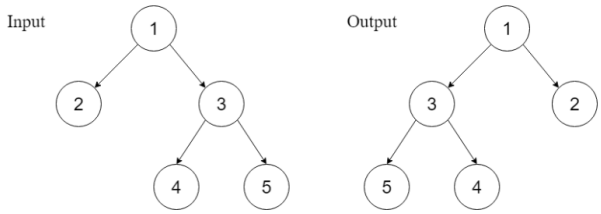

Figure 2: example1

11. [ITERATIVE PREORDER VS. POSTORDER] Given a general tree, write functions to do preorder vs. postorder traversals in an iterative fashion using an user-defined stack. Which of these traversals was a bit trickier to implement iteratively?