1 Introduction and Objective

The purpose of this tutorial is to reinforce the concepts of Binary Search Tree (BST) and the importance of having a balanced BST. In CS2040C, we learn Adelson-Velskii Landis (AVL) Tree as one such possible balanced BST implementation.

At the end of this tutorial, we will show the versatility of balanced BST data structure as alternative implementation of another ADT that we have learned earlier. We will also discuss balanced BST versus Hash Table (discussed in Tut06) as implementation for Table ADT.

2 Tutorial 07 Questions

Basic Operations of (balanced) Binary Search Tree: AVL Tree

Q1). We will start this tutorial with a quick review of basic BST operations, but on a balanced BST: AVL Tree. Tutor will first open https://visualgo.net/en/avl, click Create → Random. Then, the tutor will ask students to Search for some integers, find Successor of existing integers, perform Inorder Traversal, Insert a few random integers, and also Remove existing integers.

Q2). Draw a valid AVL Tree and nominate a vertex to be deleted such that if that vertex is deleted:
   a). No rotation happens
   b). Exactly one of the four rotation cases happens
   c). Exactly two of the four rotation cases happens (you cannot use the sample given in VisuAlgo which is https://visualgo.net/en/bst?mode=AVL&create=8,6,16,3,7,13,19,2,11,15,18,10, delete vertex 7)
Extra BST Operations

Q3). There are two important BST operations: Select and Rank that are not included in VisuAlgo yet but useful for PS4. Please discuss on how to implement these two operations efficiently.

Binary Heap... or Not?

Q4). We know that Binary (Max) Heap can be used as Priority Queue and can do ExtractMax() in $O(\log n)$ time. What modifications/additions/alterations are required so that both ExtractMax() and ExtractMin() can be done in $O(\log n)$ time for the set of $n$ elements and every other Priority Queue related-operations, especially Insert/Enqueue retains the same $O(\log n)$ running time?

Q5). Follow up from Q4). above: If you can answer Q4). before PS3 is due, would you have solved PS3 differently?

Hash Table or Balanced BST?

Q6). As of now, you have been exposed with both possible implementations of Table ADT: Hash Table (and its variations) and BST (including Balanced BST like AVL Tree). Now write down four potential usage scenarios of Table ADT. Two scenarios should favor the usage of Hash Table whereas for the other two scenarios, using Balanced BST is better.

Problem Set 4

That’s the end of Tut07. You can ask your tutor about PS4 if you encounter any difficulties with it.