COMPSCI 105: Principles of Computer Science
Summer 2007

Tutorial Eight: Tree and Heap

Helen Gu and Slobodan Vukanovic
{ygu029, svuk002}@ec.auckland.ac.nz

This tutorial is not being assessed. It provides you with an opportunity to become familiar with concepts introduced in lectures.

Question One: Binary Search Tree

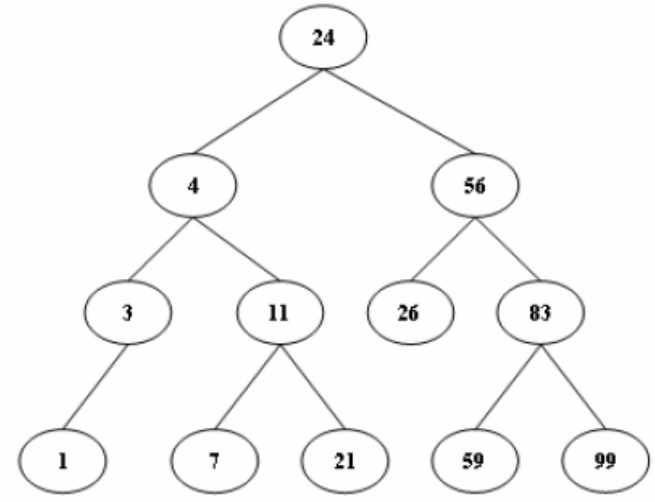
a) Show the Binary Search Tree which results after the integers 11, 24, 4, 1, 56, and 3 are added to an initially empty Binary Search Tree.

b) Now add the element 25 to the above Binary Search Tree.

c) List the elements of the tree from part b) in post-order.

d) Which node is the in-order predecessor of the root node from the tree in part b)?

e) Remove the nodes 56, 24, 21, 4 (in the order shown) from the Binary Search Tree below. Show the resulting Binary Search Tree.



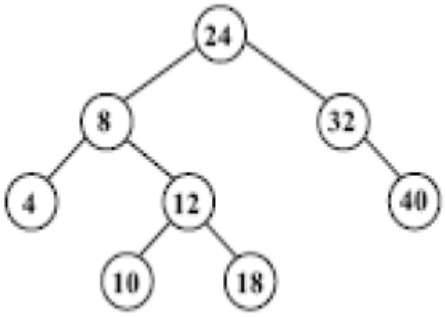
f) List the elements of your final Binary Search Tree (from part e) in pre-order.

Question Two: AVL tree

An AVL tree is a binary search tree that the heights of the left and right sub-trees on any node differ by no more than 1. After each delete or insert operation the tree is checked and if it is no longer an AVL tree then the tree is restored into an AVL tree by rotations.

a) Construct an AVL tree by inserting the items: 6, 5, 2, 1, 3 and 4 into an empty AVL Tree.

b) Draw the resulting AVL Tree after inserting 14 into the AVL Tree below:



Question Three: Heap

A heap is a complete binary tree

1. that is empty

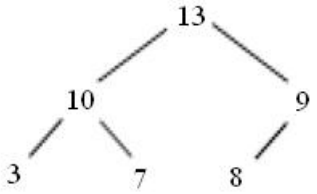
or

2. whose root contains a search key greater than or equal to the search key in each of its children, and
3. whose root has heaps as its sub-trees.

Consider Array-Based implementation of heaps.

a) Inserting the keys 2, 3, 5, 1, 6 and 4 in sequence into an initially empty heap. What does the heap look like after performed a delete operation?

b) Given the following maxheap h:



Draw the heap h after performing each of the following pseudo code operations sequentially:

i. h.heapDelete();	ii . h.heapInsert(14);	iii. h.heapInsert(12);