Computer Science 501
Data Structures & Algorithms
The College of Saint Rose
Fall 2013

Lab 8: Tree Problems

Due: 6:00 PM, Wednesday, December 4, 2013

This lab contains a number of problems related to the tree-related topics we have covered.

You may work alone or in groups of 2 or 3 on this lab. However, in order to make sure you learn the material and are well-prepared for the exam, you should work through the problems on your own before discussing them with your group members, should you choose to work in a group.

1. Consider three-node integer-valued binary trees whose nodes contain the elements "1", "2", and "3". (5 points)

   (a) Draw all valid trees whose in-order traversal would visit the nodes in the order 1-2-3. Which of these are binary search trees?

   (b) Draw all valid trees whose preorder traversal would visit the nodes in the order 1-2-3. Which of these are binary search trees?

   (c) Draw all valid trees whose postorder traversal would visit the nodes in the order 1-2-3. Which of these are binary search trees?

2. Consider a binary min-heap like the ones we have discussed in class. (5 points)

   (a) Which locations in a binary min-heap of n elements could possibly contain the third-smallest element?

   (b) Which locations in a binary min-heap of n elements could possibly contain the largest element?

3. The heaps we have discussed in class, where the heap is represented by a binary tree stored in an array, are one specific case of a more general structure called a *d-heap*. In a d-heap, each node has up to d children. So the binary heaps we have been considering would be 2-heaps. For the questions below, assume that the minimum value is stored at the root node (*i.e.*, that it is a min-heap). Note: for the parts below where you are to draw a heap, you may submit a hand drawing if you do not wish to use a drawing program. (15 points)

   (a) Draw a 2-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed?

   (b) Draw a 3-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed?

(c) For the heap element at position i in the underlying array of a 3-heap, what are the positions of its immediate chidren and its parent? (Give formulas in terms of i.)

(d) For the heap element at position i in the underlying array of a d-heap, what are the positions of its immediate chidren and its parent? (Give formulas in terms of i and d.)

(e) Draw a 1-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed?

(f) Draw a 7-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed?

4. You learned about d-heaps as part of the last question. You also learned about heapsort, which uses a 2-heap as an intermediate representation to sort the contents of an array. Let's consider a generalization of the heapsort idea:

   - First, insert the elements to be sorted into a priority queue (PQ).
   - Then, remove the elements one by one from the PQ and place them, in that order, into the sorted array.

   For heapsort, the PQ is a 2-heap, but any PQ implementation would work (naive array- or list-based with contents either sorted or unsorted, a d-heap, or even a binary search tree). Depending on which underlying PQ is used, the sorting procedure will proceed in a manner similar, in terms of the order in which comparisons occur, to one of the other sorting algorithms we have studied (*e.g.*, selection sort, quicksort, *etc.*). For each of the following underlying PQ structures, state which sorting algorithm proceeds in the manner most similar to the PQ-based sort using that PQ structure, and explain your answer briefly. (10 points)

   (a) 1-heap

   (b) 3-heap

   (c) (n-1)-heap

   (d) binary search tree

Submission

Before 6:00 PM, Wednesday, December 4, 2013, submit a PDF of your responses for grading using Submission Box at `http://sb.teresco.org` under assignment "TreeProblems".