Computer Science 431
Algorithms
The College of Saint Rose
Spring 2015

Problem Set 6: Trees

Due: 11:59 PM, Monday, March 30, 2015

You may work alone or in groups of size 2 or 3 on this assignment. Only one submission per group is needed.

Textbook Problem

? Question 1:
Levitin Exercise 5.3.2, p. 185 (3 points)

Binary Search Trees

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

? Question 2:
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? (2 points)

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

? Question 4:
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 points)

Binary Min Heaps

Consider a binary min-heap like the ones we have discussed in class.

? Question 5:
Which locations in a binary min-heap of n elements could possibly contain the third-smallest element? (2 points)

Question 6:
Which locations in a binary min-heap of n elements could possibly contain the largest element? (2 points)

Generalized Heaps

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.

Note: for each of the question below, keep in mind that the values are inserted in the order shown, *i.e.*, the first number is fully inserted into the structure before the second (and subsequent) numbers are considered at all.

Question 7:
Draw a 2-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed? (2 points)

Question 8:
Draw a 3-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed? (3 points)

Question 9:
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.) (2 points)

Question 10:
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.) (2 points)

Question 11:
Draw a 1-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed? (3 points)

Question 12:
Draw a 7-heap after the following values are inserted: 18, 9, 23, 17, 1, 43, 65, 12. How many comparisons are needed? (3 points)

Generalized Heapsort

You learned about d-heaps as you completed the questions in our previous lab. You also have 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.*).

Question 13:

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. (12 points, 3 each)

- 1-heap
- 3-heap
- (n-1)-heap
- binary search tree

Submitting

Before 11:59 PM, Monday, March 30, 2015, submit your problem set for grading. To complete the submission, upload an archive (a `.7z` or `.zip`) file containing all required files using Submission Box at `http://sb.teresco.org` under assignment "PS6".

Grading

This assignment is worth 40 points, which are distributed as follows:

Feature	Value	Score
Q1: Exercise 5.3.2	3	
Q2: in-order BSTs	2	
Q3: preorder BSTs	2	
Q4: postorder BSTs	2	
Q5: min-heap 3rd smallest	2	
Q6: min-heap largest	2	
Q7: 2-heap construction	2	
Q8: 3-heap construction	3	
Q9: 3-heap formulas	2	
Q10: d-heap formulas	2	
Q11: 1-heap construction	3	
Q12: 7-heap construction	3	
Q13: generalized heapsort	12	
Total	40	