CS140 Homework 6
Fall 2009

Instructions

1) Prepare your answers using either a word processor such as Word or Open Office or an ascii text file editor, such as vi, pico, wordpad, or emacs.
2) Print your answers and either 1) hand them in in class on Tuesday, Dec. 1, or place them in Yanjun Yao's mailbox by the final deadline of Thursday, Dec. 3 at 12 noon.

Questions

1. **(14 points)** For each of the following questions choose the best data structure to use from the below list. You may have to use the same answer for more than one question:

Hash Table	Linked List	Array
Binary Search Tree (unbalanced)	AVL Tree	
Stack	Heap	

   a. _____ You are writing a waiting list application for the door person at a popular night club. As each party of customers arrive, the door person assigns the party a priority number based on the party's perceived desirability in the club. The door person then adds them to your computerized wait list and when space becomes available, the door person admits the next party with the lowest priority number. What data structure should you use to implement the wait list?

   b. _____ You are writing an address book application. A user should be able to insert (name, address) pairs into the address book, lookup addresses by entering a name, and delete (name,address) pairs from the address book. These are the only operations that your address book must support. What data structure should you use to implement the address book?

   c. _____ You are reading input lines from a file. Each input line contains a student name and a midterm grade. With each student you want to store the set of midterms associated with that student. For each student, the midterms should be stored in the order in which they are read. You know that each student will have 3

midterms. What data structure should you use to store the midterms for each student?

d. _____ When computing taxes on selling the shares of a stock, you are allowed to select a last-in, first-out (LIFO) option, which means that the last shares that you purchased are the first ones that you consider sold. If you have a computer application that reads a set of stock transactions, what data structure should you use to store the stock transactions in order to facilitate implementing the LIFO option?

e. _____ You want to read a file that contains the names of contestants on America's Next Top Model and the number of votes that they received. You want to be able to be able to perform a variety of range queries, such as finding the top 5 contestants, the bottom 5 contests, or the number of contestants receiving between x and y votes. What data structure should you use, assuming that you feel the vote totals will be presented in a random order?

f. You are reading the names of golfers and their scores from a file. Each input line contains the name of a golfer, the golf course where the round was played, and the score for that round. For each golfer you want to store the information for each round played by that golfer. There is no limit to the number of rounds per golfer. After reading the input, the user should be able to query for a golfer and obtain the golfer's average score and a print out of the rounds played by that golfer. The insertion operation and the query operation are the only operations performed on the data.

    1. _____ What data structure should be used to store the golfers?

    2. _____ For each golfer, what data structure should be used to store the information about the golfer's rounds? You should assume that each round has one record and the rounds may be stored in any order.

2. **(4 points)** Write a one line statement that divides an integer, n, by 8 and assign the result to n. Your answer **must** use a bit expression, not division.

3. **(10 points)** Show the hash table that results if the integers 51, 21, 38, 30, 44 are inserted into the following hash table using:
    a. Linear probing, and
    b. the hash function h(k) = k % 7

    0:

    1:

    2:

    3:

    4:

    5:

    6:

4. **(10 points)** Show the hash table that results if the integers 51, 46, 23, 81, 21, 38, 30, 44, 56, and 69 are inserted into the following hash table using:
    a. separate chaining, and
    b. the hash function h(k) = k % 7

    You should append integers to the end of your lists and you should represent lists as comma separated lists of integers:

    0:

    1:

    2:

    3:

    4:

    5:

    6:

5. (**12 points**) Suppose you are given the following heap in array format:

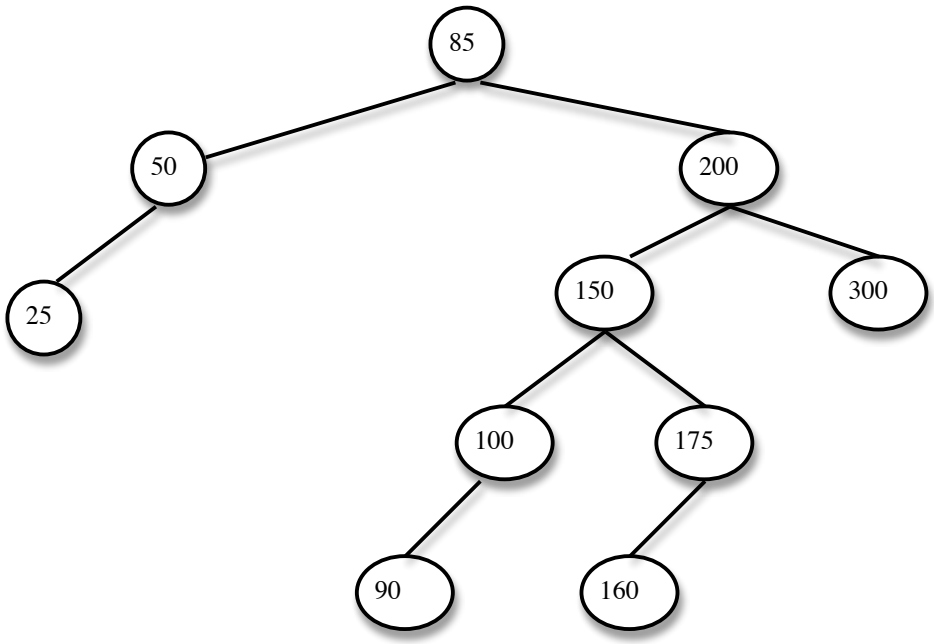
5 10 8 15 11 13 15 20 17 16 21 15

a. Draw the heap represented by this array (i.e., draw the tree representation).

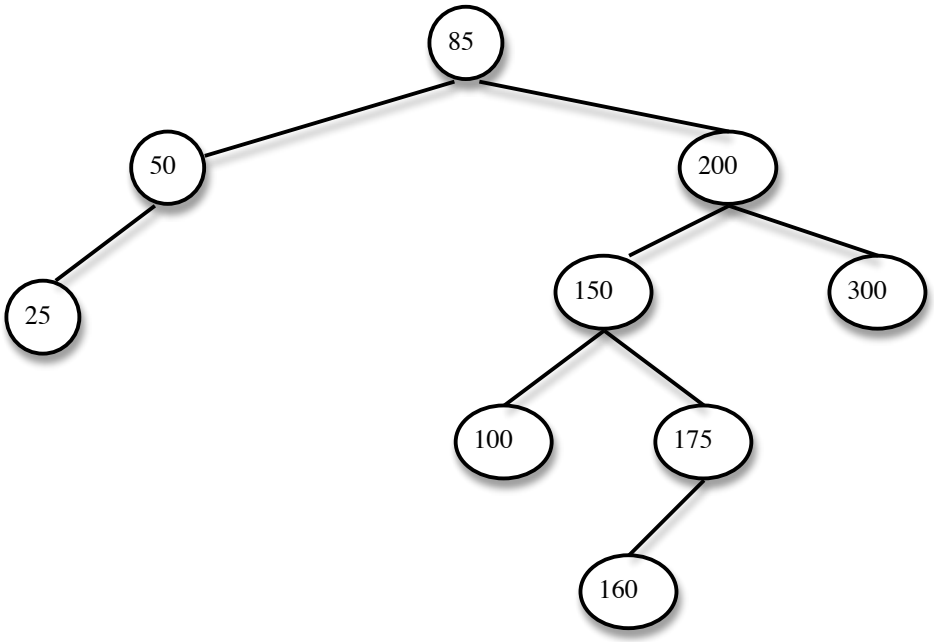
b. Show the heap that results from inserting 4. If you want partial credit then show your intermediate steps. It is ok to either draw the resulting heap or to represent it as an array.

c. Show the heap that results from performing a deleteMin on the *original* heap. If you want partial credit then show your intermediate steps. It is ok to either draw the resulting heap or to represent it as an array.

6. **(5 points)** Show the binary search tree that results if 150 is deleted from the tree below:



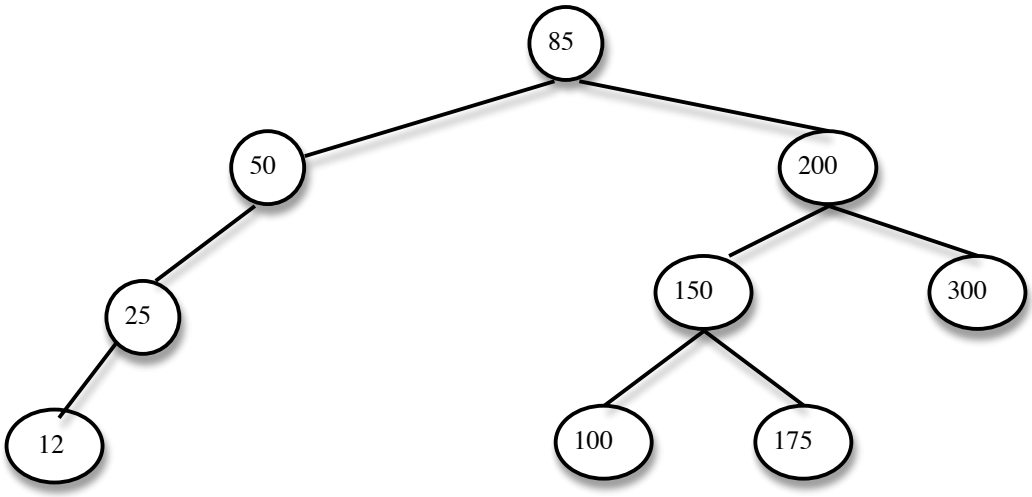
7. **(10 points)** 160 has just been inserted into the following AVL tree, causing it to violate the AVL condition:



a. Identify the **bottom-most** node that violates the AVL condition and explain why that node violates the AVL condition.

b. Use the proper rotation(s) to rebalance the above tree so that it becomes a legitimate AVL tree.

8. **(10 points)** 12 has just been inserted into the following AVL tree, causing it to violate the AVL condition:



   a. Identify the **bottom-most** node that violates the AVL condition and explain why that node violates the AVL condition.
   b. Use the proper rotation(s) to rebalance the above tree so that it becomes a legitimate AVL tree.

9. **(25 points)** Declare and write a function named **count_nodes** that counts and returns the number of nodes in a binary search tree. For example, if your function were applied to the tree in question 6, it would return 10. Assume that you have the following struct for a binary tree node:

```
typedef struct bnode {
        int key;
        struct bnode *left_child;
        struct bnode *right_child;
} BNode;
```

   Further assume that your function takes a single argument, which is a pointer to the tree's root node. **Hint:** Look at the recursive solution to the tree height problem in the previous homework. It is similar to this problem.