CS 146 St. Patrick's Day Midterm

Name:

There are 8 questions worth a total of 100 points. Please use your time wisely so you get to all of the questions. Key your answers clear.

You may use 4 pages of notes (front and back), no electronics.

Please wait to turn the page until everyone is told to begin.

1	/ 10
2	/ 10
3	/ 15
4	/ 15
5	/ 10
6	/ 10
7	/ 20
8	/ 10

100

Complexity

Question 1.

Indicate for each of the statements below whether the statement is true or false. You do not need to show your work. (Each worth 2 points.)

a) $T(N) = T(N/2) + 100$ is in $O(N)$

b) $\log \log (N + N) + \log^2 N$ is in $O(\log^2 N)$

c) buildHeap() is in O(NlogN)

d) buildHeap() is in O(N)

e) Finding the kth largest item in an AVL tree containing N elements (worst case) is in O(N)

Question 2.

Give the running time (complexity) of the statements as a function of the value of the variable N. (Each worth 5 points.) You answer should be as “tight” and “simple” as possible.

(a)

```
int sum = 0;
for (int i = 1; i < N; i *= 2)
  for (int j = 0; j < N; j++)
    sum++;
```

(b)

```
for(i = 0; i < N; i++)
  for(j = 0; j < N; j++)
     for(k = 0; k < i * j; k++)
          sum++;
```

Binary Heap Operations

Question 3.

a. Draw the binary min heap represented by the following array: (5 points)

Index	0	1	2	3	4	5
Value		1	11	13	14	15

b. Show the result of calling deleteMin twice on the heap you drew in part (a). Show the heap after each deleteMin and circle the final heap. (5 points).

c. Starting with the heap you ended up with in part (b), insert values 0 and 12 in that order. Draw the heap after each insertion and circle the final heap. (5 points)

Binary Search Tree: Question 4. (15 points)

What does the following mystery method do? Give a very short description of what the method returns (not how it does it). Assume the Node is a binary search tree and make sure this is relevant to your answer.

```
class Node {
     int value;
     Node left;
     Node right;

     static int mystery(Node n) {
          if(n==null)
               throw new IllegalArgumentException();
          if(n.left == null)
               return leftmost(n.right);
          if(n.left.left == null && n.left.right == null)
               return n.value;
          return mystery(n.left);
     }

     // return leftmost descendant of n
     static int leftmost(Node n) {
          if(n==null)
               throw new IllegalArgumentException();
          if(n.left == null)
               return n.value;
          return leftmost(n.left);
     }
}
```

Binary Tree: Question 5. (10 points)

Write pseudo-code for the following: Given a sorted (increasing order) array, write an algorithm to create a binary tree with minimal height.

AVL Trees: Question 6.

Show the AVL tree that results after each of the integer keys 22, 43, 4, 1, 2, 3, 5, and 6 are inserted, in that order, into an initially empty AVL tree. Clearly show the tree that results after each insertion, and make clear any rotations that must be performed. (10 points)

HashTable: Question 7.

Consider inserting data with integer keys 22, 14, 12, 48, 36, 32 in that order into a hash table of size 12 where the hashing function is h(key) % 12.

a) Show a chaining hash table after doing the insertions (10 points)

b) Show an open addressing with linear proving hash table after doing the insertions. (10 points)

B Trees: Question 8.

a) What is the motivation behind using a B-tree? (5 points).

b) Draw a valid B-Tree with M = 3 and L = 3. Your B-tree must have more than one internal node and more than one leaf node. (5 points)