

Name: _____

CSE 332, Spring 2010, Midterm Examination
30 April 2010

Please do not turn the page until the bell rings.

Rules:

- The exam is closed-book, closed-note. You may use a calculator *for basic arithmetic only*.
- **Please stop promptly at 3:20.**
- You can rip apart the pages, but please staple them back together before you leave.
- There are **100 points** total, distributed **unevenly** among **9** questions (many with multiple parts).

Advice:

- Read questions carefully. Understand a question before you start writing.
- **Write down thoughts and intermediate steps so you can get partial credit. But clearly circle your final answer.**
- The questions are not necessarily in order of difficulty. **Skip around.**
- If you have questions, ask.
- Relax. You are here to learn.

Name: _____

1. (8 points)

For each function below, give an asymptotic upper bound using “big-Oh” notation. Your answer should be as “tight” and “simple” as possible.

(a) $f(n) = 4n^3 + 5n^2 \log n$

(b) $f(n) = 3n + 4 \log n + 2n$

(c) $f(n) = 1.6^n + n^5$

(d) $f(n) = 2 \log \log n + \log(n/2)$

Name: _____

2. (12 points) Consider this recurrence equation:

$$\begin{aligned}T(1) &= 7 \\T(n) &= 4n + T(n-1) \quad \text{for } n > 1\end{aligned}$$

- (a) Prove that for all integers $n \geq 1$, $T(n) = 2n^2 + 2n + 3$
- (b) True or false:
- i. $T(n)$ is $O(n^2)$
 - ii. $T(n)$ is $\Omega(n^2)$
 - iii. $T(n)$ is $\Theta(n^2)$

Name: _____

3. (16 points) Suppose there is a binary min-heap with exactly 4 nodes, containing items with priorities 5, 6, 7, and 8.
- (a) Show every possible binary min-heap that could match this description. For each, draw the appropriate tree *and* the array representation.
 - (b) Argue briefly in English that your answer to part (a) is not missing anything, meaning no other binary min-heap is possible.
 - (c) For *one* of your answers to part (a), show the result of 4 successive `deleteMin` operations. Clearly indicate which heap you are starting with. You can just draw the tree (not the array).

Name: _____

4. (10 points) Give a very short description of *what* the method `mystery` 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);
    }
}
```

Name: _____

5. (12 points)

- (a) Starting with an empty AVL tree, insert the following keys in order, clearly indicating the final tree: 5, 10, 8, 4, 6, 2
- (b) Beginning with the tree produced by part (a), show the AVL tree resulting from deleting 5, then 10, then 8.

Name: _____

6. (12 points) Consider a B-Tree as defined in lecture and the textbook where $M = 64$ and $L = 6$ and the number of nodes is n . In English, describe an $O(\log_M n)$ algorithm for returning the 4th smallest element of the tree. You may assume the root is not a leaf.

Name: _____

7. (10 points) For both questions below, give an *exact* answer *and briefly justify your answer*.
- (a) What is the minimum number of data items in a B tree of height 4 (as defined in class and the textbook) with $M = 4$ and $L = 4$.
 - (b) What is the minimum number of data items in an AVL tree of height 4?

Name: _____

8. (13 points)

Consider inserting data with integer keys 7, 9, 12, 18, 29 in that order into a hash table of size 11 where the hashing function is $h(\text{key})\%11$.

- Show a chaining hash table after doing the insertions:

0	1	2	3	4	5	6	7	8	9	10

- Show an open addressing with linear probing hash table after doing the insertions.

0	1	2	3	4	5	6	7	8	9	10

- Show an open addressing with quadratic probing hash table after doing the insertions.

0	1	2	3	4	5	6	7	8	9	10

Name: _____

9. (7 points)

Consider these four implementation approaches for hash tables:

- Separate chaining
 - Open addressing with linear probing
 - Open addressing with quadratic probing
 - Open addressing with double hashing
- (a) Suppose many items for a hash table are initially hashing to the same table index. Which implementation approach or approaches are likely to lead to very poor performance?
- (b) Suppose many items for a hash table are initially hashing such that the first two hash to index i , the next two to index $i + 1$, the next two to index $i + 2$, etc. Which implementation approach or approaches are likely to lead to very poor performance?
- (c) Suppose you are trying to use various hash table libraries for some data you have, but you are getting poor performance for all of them. What should you do to try to get better hash table performance?