CS 173, Fall 2009
Midterm 2, 4 November 2009

INSTRUCTIONS (read carefully)

- Fill in your name, netid, and discussion section time below. Also write your netid on the other pages (in case they get separated).

NAME:

NETID:

DISCUSSION DAY/TIME:

- There are 5 problems, on pages numbered 1 through 5. Make sure you have a complete exam. The point value of each problem is indicated next to the problem, and in the table below.
- You have 50 minutes to finish the exam.
- Points may be deducted for solutions which are correct but excessively complicated, hard to understand, or poorly explained.
- Turn in your exam at the front. Show your ID to the proctors.
- It is wise to skim all problems and point values first, to best plan your time.
- This is a closed book exam. No notes or electronic devices of any kind are allowed. Do all work in the space provided, using the backs of sheets if necessary. See the proctor if you need more paper.
- Please bring any apparent bugs to the attention of the proctors.
- After the midterm is over, discuss its contents with other students **only** after verifying that they have also taken the exam (e.g. they aren't about to take a makeup exam).

Problem	1	2	3	4	5	total
Possible	12	10	10	10	8	50
Score						

Problem 1: Short answer (12 points)

The following require only short answers. Justification/work is not required, but may increase partial credit if your short answer is wrong.

(a) Prof. Grindelow claims that insertion sort has the same big-O running time as bubble sort. Is this correct?

(b) What is the big-O running time of Karatsuba's fast multiplication algorithm?

(c) Exactly one of the following two statements is correct. Which one is it?

  (1) For every integer p, there is an integer q such that $p \leq q$.

  (2) There is an integer q such that for every integer p, $p \leq q$.

(d) Is it true that $n!$ is $O(2^n)$?

(e) If a binary tree has height h, what is the maximum number of leaves it could have?

(f) Is the root node of a tree always an internal node?

Problem 2: Short proofs (10 points)

(a) Let $f : \mathbb{Z}^2 \to \mathbb{Z}$ be defined by $f(x, y) = xy + yx^2 - x^2$. Prove that f is onto. Do this directly from the definition of onto.

(b) Prove that $3n + 20$ is $O(n^2)$. Prove this directly from the definition of what it means for a function f to be $O(g)$ (where g is another function), being careful to justify your algebraic steps and put them into logical order.

Problem 3: Induction (10 points)

Let function $f : \mathbb{N} \to \mathbb{Z}$ be defined by

$$f(0) = 2$$

$$f(1) = 7$$

$$f(n) = f(n-1) + 2f(n-2), \text{ for } n \geq 2$$

Use strong induction on n to prove that $f(n) = 3 \cdot 2^n + (-1)^{n+1}$ for any natural number n.

Base case(s):

Inductive hypothesis:

Rest of the inductive step:

Problem 4: Algorithms (10 points)

Consider the following pseudocode for a procedure named Util:

```
Util ( a_1,..., a_n: integers)
    if (n = 1) return (a_1)^2
    else
        begin
        m = ⌈n/2⌉
        p = Util(a_1,...,a_m)
        q = Util(a_{m+1},...,a_n)
        return p + q
        end
```

(a) Explain briefly in English what Util computes.

(b) Express $T(n)$, the running time of Util on an input list of length n, as a recurrence relation. Be sure to include an initial condition. But don't worry about using floors/ceilings/etc if you need to do integer division.

(c) Give a big-theta bound on the running time of Util. Briefly explain why your answer is correct or show work (e.g. some steps from unrolling your recurrence).

Problem 5: Recurrences (8 points)

For each recurrence, check the box that correctly gives its big-O solution. Justification is not required.

Hint: is it similar to the recurrence of an algorithm you've seen? what happens if you unroll a step or two?

$T(1) = c$
$T(n) = T(n/2) + d$

$\theta(1)$: ☐ $\theta(\log n)$: ☐ $\theta(n)$: ☐ $\theta(n \log n)$: ☐

$T(1) = c$
$T(n) = T(n-1) + d$

$\theta(n)$: ☐ $\theta(n \log n)$: ☐ $\theta(n^2)$: ☐ $\theta(2^n)$: ☐

$T(1) = c$
$T(n) = 2T(n-1) + d$

$\theta(n \log n)$: ☐ $\theta(n^2)$: ☐ $\theta(2^n)$: ☐ $\theta(n!)$: ☐

$T(1) = c$
$T(n) = 2T(n/2) + dn$

$\theta(n)$: ☐ $\theta(n \log n)$: ☐ $\theta(n^2)$: ☐ $\theta(2^n)$: ☐