CS 173, Spring 2009
Midterm 2 Solutions

Problem 1: Short answer (10 points)

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

(a) Suppose $f : \mathbb{Z} \to \mathbb{Z}$ is defined by $f(n) = \lfloor \frac{n}{3} \rfloor$. Is f one-to-one?

Solution: No. For example, the inputs 0, 1, and 2 all map onto the output value 0.

(b) In a balanced binary tree of height h, is it true that all the leaves are at level h?

Solution: No. Some of the leaves may be at level $h - 1$.

(c) Is it true that $5n^3 + 17n$ is $\Omega(3^n)$?

Solution: No. 3^n grows much faster than n^3.

(d) Snape's "Unfriendly Algorithms for Wizards" textbook claims the running time of merge sort is $O(n^4)$. Is this claim correct?

Solution: Yes. This claim is technically correct, because $O(n^4)$ only gives an upper bound for how long the algorithm takes. However, it's an obnoxiously unhelpful answer, since the tight bound is $\Theta(n \log n)$.

(e) Suppose that $h : \mathbb{Z} \to \mathbb{Z}$ is defined by $h(n) = n^2 - 2$. Give a counter-example that shows that h is not onto, stating extremely briefly why it is a counter-example.

Solution: Consider the integer -327. Since n^2 cannot be negative, $h(n)$ is always at least -2. So there is no input value that maps onto -327.

Problem 2: Short answer (8 points)

(a) Recall that a **full *m-ary* tree** is a tree in which every node has either 0 or m children. If a full m-ary tree T has i internal vertices, write down an expression for the number of leaves in T. The only variables in the expression should be m and i. Briefly (in one or two sentences) explain how you arrived at your answer. (*Hint: you can use the formula you saw in lecture for the total number of vertices in a full m-ary tree.*)

Solution: A full m-ary tree with i internal vertices has a total of $mi + 1$ vertices. The number of leaves is the total number of vertices minus the number of internal nodes, which is $(mi + 1) - i = (m - 1)i + 1$

(b) Suppose that f and g are functions whose input and output values are positive real numbers. Define precisely what it means for $f(x)$ to be $O(g(x))$. (Your definition cannot use the definition of some closely-related concept such as Ω.)

Solution: $f(x)$ to be $O(g(x))$ if there are constants C and k such that $f(x) \leq Cg(x)$ for every $x > k$. (The textbook's version of the definition takes the absolute value of $f(x)$ and $g(x)$. It doesn't matter whether you used absolute values in your answer this question, because we forced $f(x)$ and $g(x)$ to be positive.)

Problem 3: Recurrences (10 points)

(a) Solve the following recurrence using unrolling. Specifically, show at least two steps of unrolling, a summation whose value is equal to $T(n)$, and finally a closed-form expression (i.e. containing no recursion or summations) equal to $T(n)$.

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 2T(n-1) + 3 \end{aligned}$$

Solution: Unrolling, two steps we get

$$T(n) = 2T(n-1) + 3 = 2(2(T(n-2)) + 6 + 3 = 2(2(2(T(n-3)))) + 12 + 6 + 3$$

In general we have

$$T(n) = 2^i T(n-i) + \sum_{k=0}^{i-1} 3(2^k)$$

We reach the base case when $i = n-1$ and have

$$T(n) = 2^{n-1} + \sum_{k=0}^{n-2} 3(2^k) = 2^{n-1} + 3(2^{n-1} - 1) = 4(2^{n-1}) - 3 = 2^{n+1} - 3$$

(b) Give a closed-form for the summation $\sum_{k=2}^{n} (\frac{1}{2})^k$

Solution: This is a finite geometric series starting from $k = 2$ rather than $k = 0$. We have

$$\sum_{k=2}^{n} (\frac{1}{2})^k = \sum_{k=0}^{n} (\frac{1}{2})^k - \sum_{i=0}^{1} (\frac{1}{2})^i = 2 \cdot (1 - (\frac{1}{2})^{n+1}) - \frac{3}{2} = (2 - \frac{1}{2^n}) - \frac{3}{2} = \frac{1}{2} - \frac{1}{2^n}$$

Problem 4: Induction (8 points)

Use induction on n to prove the following fact:

For any integer $n > 4$, $n^2 < 2^n$.

Solution:

Base case(s): If $n = 5$, then $n^2 = 25$ which is smaller than $2^n = 32$.

Inductive hypothesis: Suppose that the claim is true for $n = k$, for some $k > 4$. That is, suppose that $k^2 < 2^k$.

Rest of the inductive step: Consider $(k+1)^2$. $(k+1)^2 = k^2 + 2k + 1$. Since $k > 4$, $k^2 \geq 4k \geq 2k+1$. So $(k+1)^2 = k^2 + 2k + 1 \leq k^2 + k^2 = 2 \cdot k^2$. By the inductive hypothesis, $k^2 < 2^k$. So $2 \cdot k^2 < 2 \cdot 2^k = 2^{k+1}$. Therefore, $(k+1)^2 \leq 2 \cdot k^2 < 2^{k+1}$. So $(k+1)^2 < 2^{k+1}$, which is what we needed to show.

Because you had so little working time, I didn't penalize people who started with the goal $(k+1)^2 < 2^{k+1}$ and manipulated both sides of the equation if the comments (etc) made it clear that they understood what was given and what needed to be proved. Just be aware that I would have expected you to get your equations into a more logical order in an exam with more working time.

A common mistake was to have the main structure of the inductive step but get stuck (or have a logic gap in the middle) due to not realizing that you needed to use the fact that $2k + 1 < k^2$.

Problem 5: Algorithms (8 points)

In statistics, the *mode* is the value that occurs the most frequently in a data set. The following algorithm takes as input an arbitrary list of n integers a_1,...,a_n and attempts to find the mode of the list. The output is returned in the variable $modeValue$.

procedure FindMode(a_1,..., a_n)
$modeValue := a_1$
$modeCount := 1$
for $i := 1$ to $n - 1$
begin
    $count := 1$
    for $j := i + 1$ to n
    begin

```
            if (a_i = a_j) then
                count := count + 1
        end
        if (count > modeCount) then
        begin
            modeValue := a_i
            modeCount := count
        end
    end
    return modeValue
```

(a) Does the algorithm work correctly if the input list is $3, 3, 3, 1, 1$? What does this algorithm report as the mode?

Solution: Yes, it will return the value 3.

(b) Give a big-theta bound on the number of equality tests (i.e. in the line **if** $(a_i = a_j)$ **then**) performed by this algorithm in the worst-case. Briefly explain how you derived your answer.

Solution: Looking at the two loops, it should be clear that the outer loop will execute $n - 1$ times. Each time it executes, the inner loop will execute a diminishing number of times staring with running from 2 to n the first time and running from n to n the last time. So, we get the following sum for the number of times the equality test will execute: $(n-1) + (n-2) + ... + 2 + 1 = \sum_{k=1}^{n-1} k = \frac{(n-1)(n)}{2} = \Theta(n^2)$

Problem 6: Recursive definition (6 points)

Here is a recursive definition of a set S, which contains pairs of integers.

1. $(4, 2)$ and $(4, 1)$ are in S.
2. If (x, y) is in S, then $(x + 1, y - 1)$ is in S.
3. If (x, y) is in S, then $(x - 1, y + 1)$ is in S.

Give a non-recursive definition for the set S. Explain briefly why your answer is correct.

Solution: S contains all pairs whose coordinates add up to 5 or 6.

The two pairs given in the base case (rule 1) have this property. Neither rule 2 nor rule 3 changes the sum of the coordinates. So all pairs in S have coordinates that sum to 5 or 6.

Moreover, if we apply rules 2 and 3 repeatedly, we can change the x coordinate to any integer, positive or negative. So S contains all pairs of integers whose coordinates have the right sum.

Another approach to describing this set is geometrical. If you apply rules 2 and 3 to the point $(4, 2)$, they produce all the points with integer coordinates along the line through $(4, 2)$ with slope -1. Similarly for $(4, 1)$. So S is then the set of all points with integer coordinates along the lines with slope -1 through the points $(4, 2)$ and $(4, 1)$.