Spring 2010 CSE310 Midterm Examination 01B (in class)

Instructions:

- There are five problems in this paper. Please use the space provided (below the questions) to write the answers.
- Budget your time to solve various problems (roughly 15 minutes for each problem) and avoid spending too much time on a particular question.
- This is a **closed book** examination. You may not consult your books/notes.
- You are **NOT** supposed to use a pencil. If you use pencil, you cannot challenge your grade after the midterm is graded.

NAME	
ASUID	

Question	Score
P1	
P2	
P3	
P4	
P5	
Total	

Problem 1B. (20 points: 4 + 4 + 4 + 4 + 4)

Use O, Ω, or Θ to relate each of the following pairs of functions. In particular, you need to write your answer as $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$. In case $f(n) = \Theta(g(n))$, you will not receive credit if you write $f(n) = O(g(n))$ or $f(n) = \Omega(g(n))$, although both are implied by $f(n) = \Theta(g(n))$.

1. $f(n) = n \log n$, $g(n) = \sum_{i=1}^{n} i$.

   Solution:

   $f(n) = O(g(n))$

2. $f(n) = \sum_{i=1}^{n} 2^i, g(n) = n^{100}$.

   Solution:

   $f(n) = \Omega(g(n))$

3. $f(n) = \sum_{i=1}^{n} n(n-i), g(n) = n^3$.

   Solution:

   $f(n) = \Theta(g(n))$

4. $f(n) = n^n, g(n) = 2^n \times 3^n$.

   Solution:

   $f(n) = \Omega(g(n))$

5. $f(n) = \sum_{i=1}^{n} \frac{1}{i}, g(n) = n^{0.0001}$.

   Solution:

   $f(n) = O(g(n))$

Grading Keys:

4pts for each subproblem.

Problem 2B. (20 points: 5 + 5 + 5 + 5)

There are two algorithms A_1, A_2, with time complexities $T_1(n), T_2(n)$, respectively. We know that $T_1(n) = 64T_1(n/4) + 100n^2$; $T_2(n) = 63T_2(n/4) + 800n^2$;

(5 pts) Use the master method to decide the asymptotic notation of $T_1(n)$.

Solution:

For this recurrence, we have $a = 64$, $b = 4$, $f(n) = 100n^2$, and thus we have that $n^{\log_b a} = n^{\log_4 64} = n^3$. Since $100n^2 = O(n^{\log_b a - \epsilon})$, where $\epsilon = 0.1$, we can apply case 1 of the master method and conclude that the solution is $T_1(n) = \Theta(n^3)$.

Grading Keys:

2pts for justifying the right case;

3pts for conclusion.

(5 pts) Use the master method to decide the asymptotic notation of $T_2(n)$. You may use $\log_4 63 = 2.9886$ in your calculations.

Solution:

For this recurrence, we have $a = 63$, $b = 4$, $f(n) = 800n^2$, and thus we have that $n^{\log_b a} = n^{\log_4 63}$. Since $n^2 = O(n^{\log_4 64 - \epsilon})$, where $\epsilon = 0.1$, we can apply case 1 of the master method and conclude that the solution is $T_2(n) = \Theta(n^{\log_4 63})$.

Grading Keys:

2pts for justifying the right case;

3pts for conclusion.

(5 pts) Which algorithm is asymptotically faster?

Solution:

The algorithm A_2 is asymptotically faster.

Grading Keys:

5pts for right answer.

(5 pts) The recurrence $T(n) = 8T(n/2) + n$ describes the running time of an algorithm A. A competing algorithm A' has a running time of $T'(n) = aT'(n/4) + n$. What is the smallest integer value for a such that A' is asymptotically slower than A?

Solution: