CSE 235 Homework 04 Fall 2007

Due 1:30PM, Friday, November 9th, 2007

Name/CSE Login ______________________

Name/CSE Login ______________________

Instructions Follow instructions *carefully*, failure to do so may result in points being deducted. Clearly label each problem and submit the answers *in order*. It is highly recommended that you typeset your homework using LaTeX or a similar typesetting system. Staple this cover page to the front of a hardcopy of your assignment for easier grading. Late submissions *will not be accepted*. Be sure to show sufficient work to justify your answer(s). If you are asked to prove something, you must give as formal, rigorous, and complete proof as possible. You are to work individually, and all work should be your own. The CSE academic dishonesty policy is in effect (see `http://www.cse.unl.edu/undergrads/academic_integrity.php`).

Material (Partial Orders), Algorithms, Algorithm Analysis, Number Theory

Partner Policy You may work in pairs, but you must follow these guidelines:

1. You must work on *all* problems *together*. You may not simply partition the work between you.
2. You must use LaTeX and you may divide the typing duties however you wish.
3. You may not discuss problems with other groups or individuals.
4. Hand in only one hard copy under the first author's name.

Problem	Points	Score
A	10	
B	10	
C	8	
D	8	
E	8	
F	8	
G	8	
H	8	
3.1.24	8	
3.1.30	8	
3.2.8	8	
3.2.24ab	8	

Topics: Algorithms, Algorithm Analysis, Asymptotics.

For 3.2.24ab, use the *definitions*; do not use limits.

When asked to give an algorithm, be sure to give good pseudocode. In addition, also analyze your algorithm and give an asymptotic characterization. You may find the `algorithm2e` package useful in typesetting algorithms.

Problem A Let

$$\begin{aligned} f(n) &= (k_1)^{c_1 n} \\ g(n) &= (k_2)^{c_2 n} \end{aligned}$$

Where k_1, k_2, c_1, c_2 are all real numbers greater than 1. Under what conditions can you say that $f(n) \in \mathcal{O}(g(n))$

Problem B

(a) What is wrong with the following argument showing that

$$1^k + 2^k + \cdots + n^k \in \mathcal{O}(n^{k+1})$$

proof Let $f(n) = 1^k + 2^k + \cdots + n^k$ and $g(n) = n^{k+1}$. Then we have that

$$\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} &= \lim_{n\to\infty} \frac{1^k+2^k+\cdots+n^k}{n^{k+1}} \\
&= \lim_{n\to\infty} \frac{1}{n}\left[\frac{1^k+2^k+\cdots+n^k}{n^k}\right] \\
&= \lim_{n\to\infty} \frac{1}{n}\left[\frac{1^k}{n^k}+\frac{2^k}{n^k}+\cdots+\frac{n^k}{n^k}\right] \\
&= \lim_{n\to\infty} \frac{1}{n}\left[\left(\frac{1}{n}\right)^k+\left(\frac{2}{n}\right)^k+\cdots+\left(\frac{n}{n}\right)^k\right] \\
&= \lim_{n\to\infty} \frac{1}{n}\left[0+0+\cdots+1\right] \\
&= \lim_{n\to\infty} 0\cdot\left[0+0+\cdots+1\right] \\
&= 0
\end{aligned}$$

Therefore, $f(n) \in \mathcal{O}(g(n))$.

(b) Prove that

$$1^k + 2^k + \cdots + n^k \in \mathcal{O}(n^{k+1})$$

Problem C Order the following functions in increasing order of growth. You need not give a formal proof for each.

$$6n\log(n) + 2n, \left(\frac{1}{3}\right)^n, n^n, \log\log(n),$$
$$\log^2(n), \frac{1}{n}, 2^{10}, n - n^3 + 6n^5, \frac{n}{\log(n)}, n!,$$
$$2^{\log(n)}, 2^n, 2^4 n, 4^2 n, 3n + \log(n^{100}), \log(n)\log\log(n)$$

Problem D Give an example of a positive function $f(n)$ such that $f(n)$ is neither $\mathcal{O}(n)$ nor $\Omega(n)$ (hint: make it a discontinuous function depending on if n is odd or even).

Problem E Give and analyze an algorithm for the following problem.

Given an n vertex convex polygon described by coordinates $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, find the three vertices whose corresponding triangle has maximum perimeter.

Problem F Three or more points are *co-linear* if they lie on the same line in 2-space. Give and analyze an algorithm for the following problem.

Given a list of n points, find the maximal number of co-linear points.

Problem G What's wrong with the following about the existence of a polynomial-time algorithm to find a factor an integer n: For each integer $i = 2, \ldots \sqrt{n}$, simply check if n is divisible by i. Output the first such integer i that does divide n, otherwise return that n is prime. If division is considered our elementary operation, then clearly the algorithm is $\mathcal{O}(\sqrt{n})$ and so we have a polynomial time algorithm for factorization.

Problem H The space complexity of an algorithm is the greatest amount of memory required at any one instant *not counting the input*. What is the space complexity of Linear Search (page 170), Insertion Sort (page 174), and Algorithm 6 (page 175)?