CSE310 Homework 1 Grading Keys

Please note that you have to typeset your assignment using either LaTeX or Microsoft Word. Hand-written assignment will not be graded. You need to submit a hardcopy at the start of the lecture on the due date. You also need to submit an electronic version at the digital drop box on the blackboard. For the electronic version, you should name your file using the format CSE310-HW01-LastName-FirstName.

1. (10 pts) For each function $f(n)$ (the row index in the following table) and time t (the column index in the following table), determine the largest size n of a problem that can be solved in time t, assuming that the algorithm takes $f(n)$ seconds to solve an instance of a problem of size n. Fill the value n in the corresponding entry. Note that n has to be an integer.

	1 second	1 minute	1 hour	1 day	1 month
$\log_2 n^2$	1	$2^{\frac{60}{2}}$	$2^{\frac{60*60}{2}}$	$2^{\frac{24*60*60}{2}}$	$2^{\frac{30*24*60*60}{2}}$
$n^2 + 1000$	0	0	50	292	1609
$n^3 + 100n$	0	0	13	43	137
2^n	0	5	11	16	21

Grading: 0.5 pts for each of the 20 entries.

2. (10 pts) Let $f(n) = 100n^2 + 800n$. Prove that $f(n) \in \Theta(n^2)$.

Proof: Let $C_1 = 100, C_2 = 200, n_0 = 8$. For all $n > n_0$, we have

$$f(n) = 100n^2 + 800n \geq C_1 \times n^2 = 100n^2 > 0,$$
$$f(n) = 100n^2 + 800n \leq C_2 \times n^2 = 200n^2.$$

According to the definition, we have $f(n) \in \Theta(n^2)$ and this proves the claim.
Grading: The correctness in C_1 and C_2 is 3 pts each, and the correctness in n_0 is 2 pts for each direction.

3. (10 pts) Prove that $\Theta(n^2) + O(n^3) \subseteq O(n^3)$.

Note that for this problem, you are proving that the set of functions on the left hand side (LHS) is a subset of the set of functions on the right hand side (RHS). The set on the LHS is the algebraic sum of two sets (not the union): an element of the LHS has the form $f(n) = f_1(n) + f_2(n)$, where $f_1(n) \in \Theta(n^2)$ and $f_2(n) \in O(n^3)$.

Proof: Let $f(n)$ be any function in the set of LHS. According to the definition, there must be two functions $g(n) \in \Theta(n^2)$ and $h(n) \in O(n^3)$ such that $f(n) = g(n) + h(n)$. Since $g(n) \in \Theta(n^2)$, there are positive constants c_{g1} and c_{g2} along with big integer N_g such that

$$c_{g1} \times n^2 \leq g(n) \leq c_{g2} \times n^2 \leq c_{g_2} \times n^3$$

is true for all $n \geq N_g$. Since $h(n) \in O(n^3)$, there is a positive constant c_h along with a big integer N_h such that

$$h(n) \leq c_h \times n^3$$

is true for all $n \geq N_h$. Therefore for all $n \geq \max\{N_g, N_h\}$, we have

$$g(n) + h(n) \leq (c_{g2} + c_h) \times n^3.$$

This proves that $f(n) \in RHS$, and thus proves the claim.

Grading: 4 pts for knowing the structure that $f(n)$ can be rewritten as a summation of two functions; 2 pts for each correct upper bound, which are c_{g_2}, c_h, and $c_{g2} + c_h$, respectively.

4. (10 pts) Prove that $\Theta(n^2) + O(n^3) \neq O(n^3)$.

Proof: If we choose $f(n) = 1$, obviously, $f(n)$ is an element of RHS. However, $f(n) \notin \Theta(n^2) + O(n^3)$, since there is no positive constant c_1 and integer N such that for all $n \geq N$,

$$f(n) = 1 \geq c_1 \times n^2$$

is true. So $f(n)$ is not an element of LHS, and thus we know $\Theta(n^2) + O(n^3) \neq O(n^3)$.

This completes the proof.

Grading: 5 pts for finding the correct $f(n)$, 5 pts for correctness of the proof.

5. (10 pts) Prove that $3^n \notin \Theta(2^n)$.

Proof: Since

$$\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty} (\frac{3}{2})^n = \infty.$$

Hence, for any $C > 0$, there exists a positive integer N such that $\frac{3^n}{2^n} > C$ for all $n \geq N$ is true.

This proves $3^n \notin O(2^n)$, thus proves that $3^n \notin \Theta(2^n)$.

Grading: 5 pts for knowing $\lim_{n\to\infty} \frac{3^n}{2^n} = \infty$, 5 pts for proving it correctly.