CSCE 222-501 Discrete Structures for Computing
Fall 2014 – Hyunyoung Lee

Problem Set 2

Due dates: Electronic submission of hw2.tex and hw2.pdf files of this homework is due on **Wednesday 9/24/2014 before 11:59 p.m.** on csnet.cs.tamu.edu. A signed paper copy of the pdf file is due on **Thursday 9/25/2014** at the beginning of class. If you do not turn in the signed paper copy of the pdf file, your work will not be graded.

Name: Eric E. Gonzalez

Resources. (Discrete Mathematics and its Applications 7th Edition by Rosen)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Signature: ______________________________________

Problem 1. (10 points) Nicely typeset the definitions of the Big Oh, Big Omega, Big Theta asymptotic notations in LaTeX, **as shown in class** (that is, with the absolute values).

[Incidentally, I recommend that you keep a LaTeX file with all definitions and important theorems that we learn in this class. This will help you to memorize the definitions, and will allow you to quickly access this information when solving homework problems and studying for exams.]

Solution.
Big Oh: f $\in$ O(g) if and only if there exists some natural number n_0 and a positive real constant U such that $|f(n)| \leq$ U$|g(n)|$ for all n where n $\geq n_0$.
Big Omega: f(n) = Ω(g(n)) if and only if there exists a positive constant L and a natural number n_0 such that L$|g(n)| \leq |f(n)|$ holds for all n $\geq n_0$
Big Theta: f(n) = Θ(g(n)) if and only if there exist positive real constants L and U and a natural number n_0 such that L$|g(n)| \leq |f(n)| \leq$ U$|g(n)|$

Problem 2. (15 points) Let f_1, f_2, f_3, f_4 be functions from the set **N** of natural numbers to the set **R** of real numbers. Suppose that $f_1 = O(f_2)$ and $f_3 = O(f_4)$. Use the definition of Big Oh *given in class* to prove that

$$f_1(n) + f_3(n) = O(\max(f_2(n), f_4(n))).$$

Solution.
$|f_1(n)| \leq U_1|f_2(n)|$ holds for all $n \geq n_1$ and $|f_3(n)| \leq U_2|f_4(n)|$ for all $n \geq n_2$.
$|f_1(n)| + |f_3(n)| \leq U_1|f_2(n)| + U_2|f_4(n)|$
$\leq U_1|f(n)| + U_2|f(n)|$

$= (U_1 + U_2)|f(n)|$
$=$ U$|f(n)|$
As such, the statement $f_1(n) + f_3(n) = O(\max(f_2(n), f_4(n)))$ holds true according to Theorem 2 of Section 3.2.

Problem 3. (10 points) Let f_1, f_2, f_3 be functions from the set $\mathbf{N}$ of natural numbers to the set $\mathbf{R}$ of real numbers. Suppose that $f_1 = \Omega(f_2)$ and $f_2 = \Omega(f_3)$. Is it possible that

$$f_1(n) < f_3(n)$$

holds for all natural numbers n? Give a proof or give an argument that this is impossible.

Solution.
Let f_1 = 3n+1, f_2 = 4n+2, and f_3 = 5n + 4
f_3 - f_1 = 2n + 3
Therefore, $f_1(n) < f_3(n)$ holds for all natural numbers of n.

Problem 4. (15 points) Determine which of the following statements are correct. In each case, answer correct or incorrect, and justify your answer.
(a) $n^3 = O(n^2 + n^3)$
(b) $n^3 = O(n^2 + n \log n)$
(c) $n^3 = O(\frac{n^3}{2} + 100n^2)$

Solution.
(a) Correct. $|n^3| \leq$ U$|n^2 + n^3|$ satisfies all values n $\geq n_0$ if U=1 and n_0=1.
(b) Incorrect. There are no values for U and n_0 that satisfy $|n^3| \leq$ U$|n^2 + n^3|$ for all values n $\geq n_0$.
(c) Correct. $|n^3| \leq$ U$|n^3/2 + 100n^2|$ satisfies all values n $\geq n_0$ if U=2 and n_0=1.

Problem 5. (10 points) Prove that
(a) $n^n = \Omega(2^n)$ holds,
(b) and that $2^n = \Omega(n^n)$ does not hold.

Solution.
(a) L$(2^n) \leq n^n$ when L = 1 and n_0 = 2. $n^n \geq 2^n$ holds for all values n $\geq n_0$. Therefore, $n^n = \Omega(2^n)$ holds.
(b) $2^n \geq n^n$ does not hold for all values n $\geq n_0$. An example is that 2^3 is not $\geq 3^3$. As such, no values for constant U or $n_0 > 2$ satisfy the statement. Therefore, $2^n = \Omega(n^n)$ does not hold.

Problem 6. (15 points) Does $\Theta(n^3 + 2n + 1) = \Theta(n^3)$ hold? Justify your answer.

Solution.
Θ covers both upper and lower bounds of functions. The common upper bound value of n^3 between the two expressions links them such that n^3 = O(n^3 + $2n$ +1) and n^3 + $2n$ +1 = O(n^3). Therefore $\Theta(n^3 + 2n + 1) = \Theta(n^3)$ holds.

Problem 7. (10 points) Let k be a fixed positive integer. Show that

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

holds.

Solution.
No matter the value of k, the sum of the first n integers to the k-th power never surpasses n^{k+1}. Therefore, $|f(n)| \leq \mathrm{U}|g(n)|$ and the statement holds.

Problem 8. (15 points) Suppose that you have two algorithms A and B that solve the same problem. Algorithm A has worst case running time $T_A(n) = 2n^2 - 2n + 1$ and Algorithm B has worst case running time $T_B(n) = n^2 + n - 1$.
(a) Show that both $T_A(n)$ and $T_B(n)$ are in $O(n^2)$
(b) Show that $T_A(n) = 2n^2 + O(n)$ and $T_B(n) = n^2 + O(n)$.
(c) Explain which algorithm is preferable.

Solution.
(a) $|2n^2 - 2n + 1| \leq \mathrm{U}|n^2|$ where U=2 and n_0=1 for all n $\geq n_0$
$|n^2 + n - 1| \leq \mathrm{U}|n^2|$ where U=2 and n_0=1 for all n $\geq n_0$
Therefore, $T_A(n) \in O(n^2)$ and $T_B(n) \in O(n^2)$.

(b) 2n^2-2n+1 = 2n^2 + O(n) AND n^2 + n - 1 = n^2 + O(n)
-2n + 1 = O(n) AND n - 1 = O(n)
$|-2n + 1| \leq \mathrm{U}|n|$ where U=1 and n_0=1
$|n - 1| \leq \mathrm{U}|n|$ where U=1 and n_0=1
Therefore, $T_A(n) = 2n^2 + O(n)$ and $T_B(n) = n^2 + O(n)$.

(c) $T_B(n)$ is preferable as it has the lesser of the two worst-case running times.

Checklist:
☐ Did you add your name?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
☐ Did you sign that you followed the Aggie honor code?
☐ Did you solve all problems?
☐ Did you submit (a) your latex source file and (b) the resulting pdf file of your homework on csnet?
☐ Did you submit (c) a signed hardcopy of the pdf file in class?