Student Name: ______________________________ J#: ________________

CSC 323 Algorithm Design and Analysis, Fall 2016

Instructor: Dr. Natarajan Meghanathan

Quiz 1 (September 6, 2016) Max. Points: 30 Max. Time: 15 min.

1) (12 pts) If $t_1(n) = \Omega(g_1(n))$ and $t_2(n) = \Omega(g_2(n))$, then show that $t_1(n) + t_2(n) = \Omega(Min(g_1(n), g_2(n)))$

Given:

$t_1(n) = \Omega(g_1(n))$

i.e., $t1(n) \geq c1.g1(n)$ for some constant c and $n \geq n1$

$t_2(n) = \Omega(g_2(n))$

i.e., $t2(n) \geq c2.g2(n)$ for some constant c and $n \geq n2$

Proof:

$t_1(n) + t_2(n) \geq c1.g1(n) + c2.g2(n)$

Let there be a constant $c3 = \min(c1, c2)$; that is, $c1 \geq c3$ and $c2 \geq c3$

So, $t_1(n) + t_2(n) \geq c1.g1(n) + c2.g2(n)$

$\geq c3.g1(n) + c3.g2(n)$

$= c3 \{ g1(n) + g2(n) \}$

$g1(n) + g2(n) \geq 2*Min(g1(n), g2(n))$

So, $t_1(n) + t_2(n) \geq 2*c3*\{ Min(g1(n), g2(n)) \}$

i.e., $t_1(n) + t_2(n) \geq (constant) *\{ Min(g1(n), g2(n)) \}$

where constant = 2*c3

Hence, $t_1(n) + t_2(n) = \Omega(Min(g_1(n), g_2(n)))$

Student Name: ______________________________ J#: ________________

2) (10 pts) For a function $f(n) = \sqrt{2n^3 + 5n^2 + 3n + 2}$, find a function $g(n)$ such that $f(n) = O(g(n))$.

$f(n) = \sqrt{2n^3 + 5n^2 + 3n + 2}$. The most dominating term in this polynomial is the $\sqrt{n^3}$ term, which is $n^{3/2}$.

Let $g(n) = n^2$.

We will prove f(n) = O(g(n)) using the limit approach.

$$\lim_{n \to \infty} f(n) / g(n) = \lim_{n \to \infty} n^{3/2} / n^2 = \lim_{n \to \infty} 1/n^{1/2}. = 0.$$

Hence, f(n) = O(g(n)).

3 (8 pts) Derive the asymptotic relationship between the two functions: $n^2\log(n)$ and $n\log(n^{100})$

Let $f(n) = n^2\log(n)$ and $g(n) = n\log(n^{100}) = 100*n*\log(n)$

Note that: $\log(n^{100}) = 100*\log(n)$

$$\lim_{n \to \infty} f(n)/g(n) = \lim_{n \to \infty} n^2\log(n) / 100n\log(n) = \lim_{n \to \infty} n/100 = \infty.$$

Hence, $f(n) = \Omega(g(n))$;
That is, g(n) = O(f(n))

So, we have: $n\log(n^{100}) = O(n^2\log(n))$