University of Bristol
COMS21103: Data Structures and Algorithms
Quiz

Name: __________________________ **CS username:** ____________________

Remark: All log functions in the quiz are base 2.

Problem 1 [30 points]: Answer each of the following with either **True** or **False**, and write your answer in the corresponding box. You do **NOT** need to justify your answer:

(1) Insertion sort has the same best case and worst case time complexities. **False**

(2) Mergesort has the same best case and worst case time complexities. **True**

(3) If $f(n) = n^3 + 100n^2$, then $f(n) = \omega(n^2)$. **True**

(4) $3n^4 + o(n^4) = \Theta(n^4)$. **True**

(5) For any polynomial $p(n) = \sum_{i=0}^{d} a_i \cdot n^i$, where the a_is are constants and $a_d > 0$, we have that $p(n) = \Theta(n^d)$. **True**

(6) There exists a constant $\varepsilon > 0$, such that $n^\varepsilon = O(\log n)$. **False**

(7) $\lg(n!) = \Theta(n \lg n)$. **True**

(8) Let $f(n)$ be any asymptotically positive function. Then, $f(n) = \Theta(f(n/2))$. **False**

(9) Let $f(n)$ be any asymptotically positive function. Then, $f(n) = O\left((f(n))^2\right)$. **False**

(10) There are functions $f(n), g(n)$, for which $f(n) \neq \Omega(g(n))$ and $f(n) \neq O(g(n))$. **True**

Problem 2 [16 points]: Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \cdots, g_8$ of the functions satisfying $g_1 = O(g_2), g_2 = O(g_3), \ldots, g_7 = O(g_8)$:

$$n^{\frac{1}{\lg n}}, \quad n!, \quad 4^{\lg n}, \quad (\sqrt{2})^{\lg n}, \quad \sqrt{\lg n}, \quad \lg n, \quad 4n, \quad 2^n.$$

Answer:

$$\boxed{n^{\frac{1}{\lg n}}, \quad \sqrt{\lg n}, \quad \lg n, \quad (\sqrt{2})^{\lg n}, \quad 4n, \quad 4^{\lg n}, \quad 2^n, \quad n!}$$

Hint: One trick we need to use is to take the logarithmic function on both functions that we are going to compare. For instance, let us look at the first two functions, $n^{\frac{1}{\lg n}}$ and $\sqrt{\lg n}$. We take the logarithmic function on both functions, and obtain that

$$\lg n^{\frac{1}{\lg n}} = \frac{1}{\lg n} \cdot \lg n = 1,$$

$$\lg \sqrt{\lg n} = \lg(\lg n)^{1/2} = \frac{1}{2} \lg n.$$

Hence, $n^{\frac{1}{\lg n}} = O(\sqrt{\lg n})$.

Problem 3 [54 points]: Give asymptotic tight bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n. You only need to write your conclusion. For instance, for $T(n) = 2T(n/2) + n^3$, you only need to answer $T(n) = \Theta(n^3)$ in the corresponding box.

(1) $T(n) = 36T(n/6) + n^2$. $\boxed{T(n) = \Theta(n^2 \lg n)}$

Hint: Master Theorem, case 2.

(2) $T(n) = T(n-1) + n$ $\boxed{T(n) = \Theta(n^2)}$

Hint: By the formula of the summation $1 + 2 + \cdots + n = n(n+1)/2$ and the base case, we have that $T(n) = \Theta(n^2) + \Theta(1) = \Theta(n^2)$.

(3) $T(n) = 7T(n/3) + n^2$ $\boxed{T(n) = \Theta(n^2)}$

Hint: Master Theorem, case 3.

(4) $T(n) = T(9n/10) + n$ $\boxed{T(n) = \Theta(n)}$

Hint: Master Theorem, case 3.

(5) $T(n) = \sqrt{n}T(\sqrt{n}) + n$ $\boxed{T(n) = \Theta(n \lg \lg n)}$

Hint: We need to change the variables. See the answers of problem set 2 for a solution.

(6) $T(n) = T(n/10) + T(n/5) + T(3n/10) + n$ $\boxed{T(n) = \Theta(n)}$

Hint: Substitution methond.