Code No.: ETCS 204 Subject: Algorithm Analysis & Design

Faculty Name: Anupam Kumar Sharma **Last Date of Submission: 10/02/13**

Assignment No. 1

1. Define Algorithm.
2. Define asymptotic notations O, Ω, θ.
3. Define o and ω.
4. For inputs of size n insertion sorts runs in $8n^2$ steps, while merge sort runs in 64 nlgn. For which value of 'n' merge sort beat insertion sort?
5. What is smallest value of n such that an algorithm whose running time is 100 n^2 runs faster than an algorithm whose running time is better than 2^n?
6. Arrange following algorithmic complexities in ascending order.

   $\sqrt{n}$, 2^n, n^2, lgn, lglgn, n^n, n!
7. Show that for any real constant's a & b, where b > 0

   $(n + a)^b = \theta (n^b)$
8. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?
9. Prove that $O(g(n)) \cap \omega(g(n))$ is an empty set.
10. f(n) and g(n) be asymptotically non-negative functions. Prove max (f(n), g(n)) = θ (f(n) + g(n)).
11. Prove that $n! = \omega(2^n)$ and $n! = o(n^n)$
12. Solve the following recurrences using master method:
    a. $T(n) = 4\,T(n/2) + n$
    b. $T(n) = 4\,T(n/2) + n^2$
    c. $T(n) = 2T(n/2) + n^3$
    d. $T(n) = T(9n/10) + n$
    e. $T(n) = 16T(n/4) + n^2$
    f. $T(n) = 7T(n/3) + n^2$
    g. $T(n) = 7T(n/2) + n^2$
    h. $T(n) = 2T(n/4) + \sqrt{n}$

13. The recurrence $T(n) = 7T(n/2) + n^2$ describes the running time of an algorithm *A*. A competing algorithm *A′* has a running time of $T'(n) = aT'(n/4) + n^2$. What is the largest integer value for *a* such that *A′* is asymptotically faster than *A*?

14. Solve following recurrences by substitution

a. $T(n) = T(n/2) + 1$
b. $T(n) = T(n-1) + n^k$
c. $T(n) = 2T(n/2) + n$
d. $T(n) = 2T(n/2 + 17) + n.$
d. $T(n) = T(\sqrt{n}) + 1$
e. $T(n) = T(\sqrt{n}) + \lg n$