United College of Engineering and Research, Prayagraj

Department of Computer Science and Information Technology

**Design and Analysis of Algorithm (KCS-503)**

**Assignment-1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q. No.** | **Question** | **CO** | **Bloom’s level** |
|  | Let f(n) and g(n) be asymptotically non-negative functions. Using the basic definition of‚θ-notation, prove that **max (f(n), g(n)) = θ(f(n)+g(n)).** | CO1 | L4 |
|  | Solve the followings:-  (a) Is 2n+1 = O(2n) ?  (b) Is 22n = O(2n) ? | CO1 | L3 |
|  | Take the following list of functions and arrange them in ascending order of growth rate. That is, if function g(n) immediately follows function f(n) in your list, then it should be the case that f(n) is O(g(n)). f1(n) = n2.5, f2(n) = √2n, f3(n) = n + 10, f4(n) = 10n, f5(n) = 100n, and f6(n) = n2 log n | CO1 | L4 |
|  | Rank the following by growth rate: n, 2lg √n, log n, log (logn), log2n, (lgn)lgn, 4, (3/2)n, n! . | CO1 | L3 |
|  | How will you sort following array A of elements using heap sort: A = (23, 9, 18, 45, 5, 9, 1, 17, 6). | CO1 | L3 |
|  | Solve the recurrence T (n) = 2T(n/2) + n2+ 2n+ 1 | CO1 | L3 |
|  | Solve the recurrence using recursion tree method:  T (n) = T (n/2) + T (n/4) + T (n/8) + n | CO1 | L4 |
|  | The recurrence T (n) = 7T (n/3) + n2 describes the running time of an algorithm A. Another competing algorithm B has a running time of S (n) = a S (n/ 9) + n2. What is the smallest value of ‘a’ such that A is asymptotically faster than B? | CO1 | L4 |
|  | Solve the recurrence relation by substitution method  T(n)= 2T(n/2) + n | CO1 | L3 |
|  | Find the time complexity of the recurrence relation  T(n) = n +T(n/5)+T(7n/10) | CO1 | L3 |