ESc 101: ALGORITHMS AND PROGRAMMING

Y. Narahari
Indian Institute of Science
Bangalore, India
August 17, 2011

Problem Set 1

1. Which of these functions grows faster?
   (a) $n^{\log n}$; $(\log n)^n$
   (b) $\log n^k$; $(\log n)^k$
   (c) n^n; $n!$.
2. Prove the following statements.
   (a) $\frac{n(n-1)}{2}$ is $O(n^2)$
   (b) $\sum_{i=1}^{n} i^k$ is $O(n^{k+1})$, for integer k
3. Given an array with n unsorted, integer elements, what is the worst case complexity of the following problems in terms of n: (a) Finding the maximum element (b) Finding the first largest and second largest elements (c) Finding the mean (d) Finding the median (e) Finding the standard deviation (f) Finding the mode (g) Removing all duplicates (h) Reversing the elements of the array (i) Partitioning the array with respect to a given pivot element so that all elements less than the pivot appear before all elements greater than or equal to the pivot.
4. Given square matrices with n rows and n columns, what is the worst case computational complexity of the following matrix algorithms in terms of n: (a) Matrix Addition (b) Matrix Multiplication (c) Finding if the given matrix is a zero matrix (d) Finding if the given matrix is a symmetric matrix (e) Computing the determinant of the given matrix (f) Transposing the matrix in situ (in place).
5. If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$ where f_1 and f_2 are positive functions of n, show that the function $f_1(n) + f_2(n)$ is $O(\max(g_1(n), g_2(n)))$.
6. Solve the following recurrences, where $T(1) = 1$ and $T(n)$ for $n \geq 2$ satisfies:
   (a) $T(n) = T(\frac{n}{2}) + 1$
   (b) $T(n) = 2T(\frac{n}{2}) + n^2$
   (c) $T(n) = 2T(n-1) + 1$
   (d) $T(n) = 2T(n-1) + n$
7. Prove or disprove: 3^n is $O(2^n)$.
8. An array has integer elements in random order. Think of an efficient algorithm to rearrange the elements of the array so that all odd numbers will appear first followed by all even numbers.