ASYMPTOTIC ANALYSIS

(1) Order the following functions in the increasing order of their asymptotic growth rate: (a) $4n \log n + 2n$, (b) 2^{10}, (c) $2^{\log n}$, (d) $3n + 100 \log n$, (e) $4n$, (f) 2^n, (g) $n2 - 10n$, (h) n^3, (i) $n \log n$

(2) Show that $O(\max\{f(n), g(n)\})$ is $O(f(n) + g(n))$.

(3) Show that if $d(n)$ is $O(f(n))$ and $e(n)$ is $O(g(n))$ then $d(n) - e(n)$ is not necessarily $O(f(n) - g(n))$.

(4) Show that n^2 is not $o(n^2)$.

(5) Suppose that each row of an $n \times n$ array A consists of 1s and 0s such that, in any row of A, all the 1s come before any 0s in that row. Assuming A is already in memory, describe a method running in $\Theta(n)$ time for finding the row of A that contains the most 1s. Explain why your algorithm is $\Theta(n)$.

(6) Describe in pseudo-code a method for multiplying an $n \times m$ matrix A and an $m \times p$ matrix B. Recall that the product $C = AB$ is defined so that $C[i][j] = \sum_{k=1}^{m} A[i][k] \cdot B[k][j]$. What is the running time of your method?

(7) [Challenge Problem] Describe a method for finding both the minimum and maximum of n numbers using fewer than $3n/2$ comparisons.