CSE 101 Calibration Homework

Background, (Order and Recurrence Relations, simple algorithms), MergeSort

Due October 3

100 points total, DOES NOT COUNT TOWARDS GRADE

Is $4^\log n \in O(n^2)$? Why or why not? (When unspecified, logs are base 2). Is $\log(n!) \in \Omega(n \log n)$? Why or why not? Is $4^n \in O(2^n)$? Why or why not? Is $n + (n - 1) + (n - 2) + \ldots + 1 \in O(n)$? Why or why not? (5 points each.)

Page 496-501 of text; 10 pts.each 1e, 6

Triangles (20 points) Let $G$ be an undirected graph with nodes $v_1, \ldots, v_n$. The adjacency matrix representation for $G$ is the $n \times n$ matrix $M$ given by: $M_{ij} = 1$ if there is an edge from $v_i$ to $v_j$, and $M_{ij} = 0$. A triangle is a set $\{v_i, v_j, v_k\}$ of 3 distinct vertices so that there is an edge from $v_i$ to $v_j$, another from $v_j$ to $v_k$ and a third from $v_k$ to $v_i$. Give and analyze an algorithm for counting the number of triangles in a graph $G$, given by its adjacency matrix representation. Analyze your algorithm's worst-case performance first in terms of just the number of nodes $n$ of the graph, then in terms of $n$ and the number of edges $m$ of the graph. Your algorithm should be faster when $m << n^2$.

Summing triples (20 points) Let $A[1, \ldots, n]$ be an array of positive integers.

A summing triple in $A$ is 3 distinct indices $1 \leq i, j, k \leq n$ so that $A[i] + A[j] = A[k]$. Give and analyze an algorithm that, given $A$, determines whether there is any summing triple in $A$. Try to be better than $O(n^3)$.

Implementation (20 points) Implement the algorithm you gave for the summing triples problem above. Try it on random arrays where each element $A[i]$ is chosen in the range $1, \ldots, n$, for $n = 2^0, 2^1, 2^1, 2^2, 2^2, 2^4, 2^6, 2^8$ and $2^{10}$. Plot its performance on a $\log_{2} n$ vs. $\log_{2}$ of the time scale. Then try the same experiment on random arrays where each element is chosen in the range $1, \ldots, n^2$. Do you see a difference? If so, can you explain it?