

CSCI 3212 Homework1

This homework is due by the beginning of class Tuesday Sept. 12. It will be accepted for full credit if uploaded to blackboard by 5pm Tuesday Sept. 12. It will be accepted with a 20% deduction if uploaded by 9am Wednesday Sept. 13, after which it will not be accepted.

1. Rank the following functions in increasing order of asymptotic growth rate. For functions that are asymptotically equivalent, group them together. You do not need to prove your ordering, but you may supply explanations for the purpose of getting partial credit.

$$\begin{aligned}f_1(n) &= 500 n^3 \\f_2(n) &= 17n + (2/n^2) \\f_3(n) &= 7n \lg \lg n \\f_4(n) &= 20n \lg^3 n + 5n^2 \\f_5(n) &= 4\sqrt{n} + 3\lg(n^2) \\f_6(n) &= 2^{\sqrt{n}} \\f_7(n) &= 2^{(3\lg n)} \\f_8(n) &= 5 \lg^2 n + 10\lg n \\f_9(n) &= 20 \lg(n^2) \\f_{10}(n) &= 4 \log_3 n\end{aligned}$$

2. Describe an $O(n \log n)$ divide-and-conquer algorithm for the 2-d maxima problem. The input to your algorithm should be an array of n points $P[1..n]$, where $P[i].x$ and $P[i].y$ are the x - and y -coordinates of the i th point. You may store the output however you like. Your algorithm should not explicitly invoke any sorting algorithms. Informally explain your algorithm's correctness, and derive its asymptotic running time. You may assume that there are no duplicate x - or y -coordinates, if it helps simplify your algorithm.
3. You are given an array $A[1..n]$ of real numbers, some positive some negative. Design an $O(n \log n)$ algorithm which determines whether A contains two elements $A[i]$ and $A[j]$ such that $A[i] = A[j]$. (If A contains the element 0, then the answer is always yes.) Briefly explain your algorithm and derive its running time.