Algorithm analysis, asymptotic notation

September 26, 2009

Homework 1

Due Date: Tuesday, 6 October 2009 by end of lecture

Problem 1-1. (15pts) True/False: Can you say that the function $f(n) = 2^{\sqrt{\log n}}$ is:
$O(n)$? $O(n^2)$? $O(\log n)$? $o(n)$? $o(\log n)$? $\Omega(n)$? $\Omega(1)$? $\Omega(\log n)$? $\omega(n)$? $\omega(1)$? $\Theta(n)$? $\Theta(n^2)$? $\Theta(\log n)$?

Problem 1-2. (15pts) Consider an instance of the Stable Matching problem discussed in class, in which there exists a man m and a woman w such that m is ranked first on the preference list of w and w is ranked first on the preference list of m. Is it true that in every stable matching for this instance, the pair (m, w) are always matched ? Prove that the statement is correct or give a counterexample. Assume all preferences are distinct, i.e. there are no two men that a woman prefers equally or two women that a man prefers equally.

Problem 1-3. (15pts) Given an array A of n elements, you need to produce a two-dimensional array B where $B[i, j] = \sum_{k=i}^{j} A[k]$.

1. What is the running time of the algorithm that computes B by (brute-force) using the above formula ?
2. Describe an algorithm that is asymptotically faster, i.e. its running time is small-oh of the running time of the brute-force algorithm.

Problem 1-4. (15pts) Describe (in precise, unambiguous English) an algorithm which takes two unordered sets as input: set A, with n elements and set B, with m elements and returns the intersection of the these two sets. Assume that the sets are given to you in 2 arrays, each element represented by an integer, with different integers representing different elements. "Unordered" in this context means that the integers in the arrays are not necessarily in sorted order. [The brute-force $O(mn)$ algorithm will get no points.] What can you say about the asymptotic running time of your algorithm ? State any assumptions you make about the data structures you use. How will your algorithm and the asymptotic behavior change (if at all) if the input is given to you as 2 linked lists ?

Problem 1-5. (15pts) Show that for any positive constants a and b,

$$(n + a)^b = \Theta(n^b)$$

Is the above claim still true when a is allowed to be negative ? (Prove that it is true, or give a counterexample.)

Problem 1-6. (15pts) Explain why statement "*Worst-case running time of algorithm A is at least* $O(n^2)$" is meaningless.

Problem 1-7. (Extra Credit) Given a sorted array of n numbers, describe a $\Theta(n)$-algorithm that, given another number x, determines whether or not there exist two elements in in the input array whose sum is exactly x.