Basic Algorithms CSCI-UA.0310

Sample Test

Problem 1 (30 points)

Verify if the following statements are true or false. Fully justify your answers.

(a) If $f = O(g)$ and $h = O(g)$, then $f = \Omega(h)$.

(b) If $f = \log_2(n!)$ and $g = \log_2(n^n)$, then $f = \Theta(g)$.

Hint: Note that $n! < n^n$ and $\left(\frac{n}{2}\right)^{\frac{n}{2}} < n!$ for $n \geq 3$.

(c) If $T(n) = T(\frac{2n}{5}) + T(\frac{3n}{5}) + n$, then we have $T(n) = \Theta(n \log n)$.
You may assume that n is a power of 5, i.e., $n = 5^k$ for some positive integer k.

Problem 2 (35 points)

Let $A[1 \ldots n]$ be an array of length n consisting of some positive and negative integers.

(a) Use divide and conquer to devise an algorithm that outputs the elements of A in such a way that the positive elements follow negative elements while maintaining the relative ordering of the elements in A. Your algorithm must work in $\Theta(n \log n)$ time.

Example: If $A = [-2, 10, 5, -8, -4, -10, 12, 1]$, then the output must be $[-2, -8, -4, -10, 10, 5, 12, 1]$. Note that the relative ordering of the negative elements and the relative ordering of the positive elements are preserved.

Hint: Modify MergeSort!

(b) Draw the corresponding recursion tree for your algorithm and justify that the running time of your algorithm is $\Theta(n \log n)$.

(c) Can you devise an algorithm with time complexity $\Theta(n)$? Justify your answer.

Problem 3 (35 points)

You have an $m \times n$ chocolate bar. You are also given a two-dimensional array of prices $P[1 \ldots m][1 \ldots n]$, where $P[i][j]$ denotes the price of the $i \times j$ chocolate bar for $i = 1, \ldots, m$, and $j = 1, \ldots, n$.
You are allowed to repeat the following process any number of times, starting initially with the original $m \times n$ piece of chocolate bar you have. In each step, you may take one of the pieces you have and split it into two pieces by cutting it either vertically or horizontally.
For example, for $m = 6$ and $n = 3$, you may first choose to horizontally split the original 6×3 bar you have into two pieces of sizes 4×3 and 2×3. Then, in the next step, you may take the 4×3 piece and vertically split it into two pieces 4×2 and 4×1. If you decide to stop, you have three pieces of sizes 2×3, 4×2, and 4×1, which you can sell for $P[2, 3] + P[4, 2] + P[4, 1]$.
The goal is to find a splitting that maximizes your total selling price.

(a) Let $S(i, j)$ be the maximum selling price you can get by splitting an $i \times j$ piece, for $i = 0, \ldots, m$, and $j = 0, \ldots, n$. For the base cases, we have $S(i, 0) = S(0, j) = 0$.
Write a recursion that $S(m, n)$ satisfies. Justify your answer.

Hint: Extend the idea of the rod cutting problem.

(b) Write a bottom-up dynamic programming algorithm to compute $S(m, n)$.

(c) Compare the time complexity of your algorithm with the time complexity of the longest common subsequence problem. Justify your answer.