Homework Assignment #2

Posted on Sunday, 4/10/2016.
Due 10PM, Monday, 4/18/2016.

1. (15 points) Suppose that you are choosing between 3 algorithms A, B, and C for your algorithmic problem:

   (a) Algorithm A solves the problem with an input of size n by dividing the problem into 5 sub-problems of half the original input size in linear time, recursively solving the sub-problems, and then combining the solutions in linear time.
   (b) Algorithm B solves the problem with an input of size n by creating two sub-problems of input size n – 1 in constant time, recursively solving the two sub-problems, and then combining the solutions in constant time.
   (c) Algorithm C solves the problem with an input of size n by dividing the problem into 9 sub-problems of input size n/3 in cubic time, recursively solving the sub-problems, and then combining the solutions in quadratic time.

   What are the time complexities of these algorithms, and which one would you prefer? Prove the time complexities.

2. (10 points) Exercise 4.3-9 in the textbook. Change the equation to the following. Prove your solution.

$$T(n) = 3 \cdot T(n^{\frac{1}{3}}) + 24 \cdot T(n^{\frac{1}{6}}) + (\log n)^2 \cdot (\log\log n)^{1.5} + \ 101{,}000{,}078$$

3. (30 points) Solve the following recurrences. Your solutions should be asymptotically tight. Do not worry about whether values are integral. Prove your solutions.

   In class, we have discussed one version of the Master Theorem in detail and mentioned the Akra-Bazzi Master Theorem. The textbook gives another version of the Master Theorem. Apply all three theorems to each of the following recurrence equations.

   (a) $T(n) = 5 \cdot T\left(\frac{n}{2}\right) + 176 \cdot T\left(\frac{n}{4}\right) + n^4(\log n)^{-3}$
   (b) $T(n) = 5 \cdot T\left(\frac{n}{3}\right) + 4 \cdot T\left(\frac{n}{6}\right) + 26 \cdot T\left(\frac{n}{9}\right) + n^2$
   (c) $T(n) = 25 \cdot T\left(\frac{n}{5}\right) + n^2 \log n$
   (d) $T(n) = 5 \cdot T\left(\frac{n}{5}\right) + n^2 \log n$

4. (15 points) Generalize Exercise 9.3-8 in the textbook by changing the input from two arrays X and Y to five arrays A, B, C, D, E each containing n numbers already in sorted order. Prove the time complexity and correctness of your algorithm.

5. (15 points) Consider a strictly decreasing function $f: \mathrm{N} \rightarrow \mathrm{Z}$ (that is, a function defined on the natural numbers and taking integer values, such that $f(i) > f(i + 1)$). Assuming we can evaluate f at any i in constant time, we want to find $n = \min\{i \in \mathrm{N} \mid f(i) \leq 0\}$ (that is, we want to find the point where f becomes non-positive).

   We can easily solve the problem in $O(n)$ time by evaluating $f(1), f(2), f(3), \ldots, f(n)$. Give an $O(\log n)$-time algorithm. Prove the time complexity and correctness of your algorithm.

6. (15 points) The our-secret-name problem (OSNP) is defined as follows. Given an array $A[1..n]$ of integers, find values of i and j with $1 \leq i \leq j \leq n$ such that

$$\left(\sum_{k=i}^{j} A(k)\right)^2$$

   is maximized.

   Give an $O(n \log n)$-time divide-and-conquer algorithm to solve OSNP. Prove the time complexity and correctness of your algorithm.