CSE 331: Introduction to Algorithm Analysis and Design Fall 2011

Homework 3

Due Friday, September 30, 2011 by 1:15pm in class

IMPORTANT: Please submit each problem separately, i.e. each problem should begin on a new page and only the pages for one problem should be stapled together. Failure to do so might result in some problem(s) not being graded.

For general homework policies and our suggestions, please see the policy document.

For this homework, you can assume that addition, subtraction, multiplication and division of two numbers can be done in $O(1)$ time. (Side question: Is this assumption justified?)

No collaboration is allowed on the first problem.

---

1. (40 points) (**You must work on this problem on your own: NO collaboration is allowed**)

   A *forest* with c *components* is a graph that is the union of c disjoint trees. Note that a tree is a forest with 1 component. Prove that an n-vertex forest with c components has $n - c$ edges. (Recall that we have proved the statement above for $c = 1$ in class.)

   (*Hint*: The above can be proven by induction. Of course, any other correct proof is also welcome).

2. ($30 + 15 = 45$ points) You should have seen the problem of representing a number in a given base in an intro CSE course. For example, base 2 representation (aka binary numbers) is the most common representation on computers though sometimes base 16 (or hex representation) is also used. (Don Knuth also writes his checks in hex.) Base 10 (or decimal representation) is more common in our daily lives. In this problem you will analyze algorithm(s) to convert a number represented in any given base into its decimal representation.

   Recall that the decimal number corresponding to the base b (for some integer $b \geq 2$) representation $(a_0, \ldots, a_{n-1})$ (for some integer $n \geq 1$, where for every $0 \leq i < n$, $0 \leq a_i \leq b-1$) is defined as the number: $\sum_{i=0}^{n-1} a_i \cdot b^i$ (all the operations are over the decimal representation). For example, the number corresponding to $(1, 2, 3)$ in base 4 representation is $1 + 2 \cdot 4 + 3 \cdot 4^2 = 57$.

   For concreteness, given an integer $b \geq 2$ and n integers $a_0, \ldots, a_{n-1}$ with $0 \leq a_i \leq b-1$ for every $0 \leq i < n$, let us denote the number $\sum_{i=0}^{n-1} a_i \cdot b^i$ by $\mathcal{N}(a_0, \ldots, a_{n-1}; b)$.

   Consider the following algorithm for for computing $\mathcal{N}(a_0, \ldots, a_{n-1}; b)$ given $a_0, \ldots, a_{n-1}, b$:

   /* Input: Integer $b \geq 2$ and integers $a_0, \ldots, a_{n-1}$[1] */

   (a) `eval` $\leftarrow 0$.

   (b) For $i = 0 \cdots n-1$

      i. `temp` $\leftarrow 1$.

[1] You can assume that each a_i satisfies $0 \leq a_i \leq b-1$.

ii. For $j = 1 \cdots i$

- $\texttt{temp} \leftarrow \texttt{temp} \cdot b$

iii. $\texttt{eval} \leftarrow \texttt{eval} + a_i \cdot \texttt{temp}$

(c) Return `eval`

Convince yourself that the algorithm above does indeed output $\mathcal{N}(a_0, \ldots, a_{n-1}; b)$. Now consider the following questions:

- Prove that the algorithm above has a running time of $\Theta(n^2)$.
- The representation fairy tells you that there is a faster algorithm for the conversion to decimal representation problem than the one above.[2] Design an algorithm with a running time of $O(f(n))$ such that $\lim_{n\to\infty} \frac{f(n)}{n^2} = 0$. You will receive more credit the (asymptotically) smaller your function $f(n)$ is.

3. ($7 + 8 = 15$ points) Exercise 8 in Chapter 2.

[2]The representation fairy never lies.