Fundamental Algorithms, Assignment 4

Due Feb 15 in Recitation

> Ever since that night by the river, when Deeti had come to his help, Kalua had kept count of the days on which he was granted a glimpse of her, and the empty days in between. The tally was kept neither with any specific intension, nor as an expression of hope – for Kalua knew full well that between her and himself, none but the most tenuous connection could exist – yet the patient enumeration happened in his head whether he liked it or not; he was powerless to make it cease, for his mind, slow and plodding in some respects, had a way of seeking the safety of numbers.
>
> Amitav Ghosh, *Sea of Poppies*

When asked for the asymptotics answer in a form $\Theta(n^a)$ or $\Theta(\lg^b n)$ or $\Theta(n^a \lg^b n)$ for some reals a, b.

1. Consider the recursion $T(n) = 9T(n/3) + n^2$ with initial value $T(1) = 1$. Calculate the *precise* values of $T(3), T(9), T(27), T(81), T(243)$. Make a good (and correct) guess as to the general formula for $T(3^i)$ and write this as $T(n)$. (Don't worry about when n is not a power of three.) Now use the Master Theorem to give, in Thetaland, the asymptotics of $T(n)$. Check that the two answers are consistent.

2. Use the Master Theorem to give, in Thetaland, the asymptotics of these recursions:

   (a) $T(n) = 6T(n/2) + n\sqrt{n}$

   (b) $T(n) = 4T(n/2) + n^5$

   (c) $T(n) = 4T(n/2) + 7n^2 + 2n + 1$

3. `Toom-3` is an algorithm similar to the Karatsuba algorithm discussed in class. (Don't worry how `Toom-3` really works, we just want an analysis given the information below.) It multiplies two n digit numbers by making five recursive calls to multiplication of two $n/3$ digit numbers plus thirty additions and subtractions. Each of the additions and subtractions take time $O(n)$. Give the recursion for the time $T(n)$ for `Toom-3` and use the Master Theorem to find the asymptotics of $T(n)$. Compare with the time $\Theta(n^{\log_2 3})$ of Karatsuba. Which is faster when n is large?

4. Write the following sums in the form $\Theta(g(n))$ with $g(n)$ one of the standard functions. In each case give reasonable (they needn't be optimal) positive c_1, c_2 so that the sum is between $c_1 g(n)$ and $c_2 g(n)$ for n large.

   (a) $n^2 + (n+1)^2 + \ldots + (2n)^2$

   (b) $\lg^2(1) + \lg^2(2) + \ldots + \lg^2(n)$

   (c) $1^3 + \ldots + n^3$.

5. Give an algorithm for subtracting two n-digit decimal numbers. The numbers will be inputted as $A[0 \cdots N]$ and $B[0 \cdots N]$ and the output should be $C[0 \cdots N]$. (Assume that the result will be nonnegative.) How long does your algorithm take, expressing your answer in one of the standard $\Theta(g(n))$ forms.

China is a sleeping giant. Let her sleep, for when she wakes she will move the world.
– Napoleon Bonaparte 1769-1821