Basic Algorithms (V22.0310); Fall 2005; Yap

QUIZ TWO (with SOLUTION)

Dec 5, 2005

Instructions

- Please read questions carefully. When in doubt, ask!
- Answer all questions.
- Write answers ONLY on the right-hand side of every double page. Use the left-hand side for your rough work.

Question 1 (30 Points)

Let $T(n) = 3T(n/9) + n$.
(a) Guess the formula for $T(n)$ after expanding the recurrence for $i \geq 1$ times.
(b) Verify your formula.
(c) Solve for $T(n)$ by choosing a value of i to terminate expansion.

SOLUTION:

(a) GUESS: $T(n) = 3^i T(n/9^i) + n\sum_{j=0}^{i-1} 3^{-i}$.

(b) VERIFY: True for $i = 1$ For $i > 1$, $T(n) = 3^i[3T(n/9^{i+1}) + n/9^i] + n\sum_{j=0}^{i-1} 3^{-i} = 3^{i+1}T(n/9^{i+1}) + n\sum_{j=0}^{i} 3^{-i}$.

(c) Choose $i = \lceil \log_9 n \rceil$. Then $n/9^i < 1$. Assume $T(x) = 0$ for $x < 1$. Hence $T(n) = n\sum_{j=0}^{i-1} 3^{-i}$. But $\sum_{j=0}^{i-1} 3^{-i} = \Theta(1)$, so $T(n) = \Theta(n)$.

Question 2 (30 Points)

Consider the recurrence $T(n) = 2T(n/2) + T(n/3) + n$. This is not in the form of a Master recurrence.

(a) Prove by induction that $T(n) = O(n^2)$.

(b) Find another function $T_1(n)$ that *is* a Master recurrence, such that $T(n) \leq T_1(n)$ (assuming they have the same initial conditions).

(c) State the bound for $T_1(n)$ using the Master Theorem.

SOLUTION:

(a) Assume $T(n) \leq Kn^2$.

$$\begin{aligned} T(n) &= 2T(n/2) + T(n/3) + n \\ &\leq 2[K(n/2)^2] + [K(n/3)^2] + n \\ &= Kn^2\left[\frac{1}{2} + \frac{1}{9} + \frac{1}{Kn}\right] \\ &\leq Kn^2 \end{aligned}$$

provided $\frac{1}{2} + \frac{1}{9} + \frac{1}{Kn} \leq 1$. But this is true provided $1/(Kn) \leq 4/9$ or $Kn \geq 9/4$. For any choice of $K > 0$, this will be true for n large enough.

(b) $T_1(n) = 3T_1(n/2) + n$ is a master recurrence and clearly, $T(n) \leq T_1(n)$.

(c) $T_1(n) = \Theta(n^{\lg 3})$.

Question 3 (30 Points)

The new Computer Science PhD's hired by Google Inc were having a competition among themselves. They want to make sure that when you type "search", then Google will ask if you want "google", among a list of alternatives to the word "search". Of course, they want to do this in a principled way.

In general, Google Search offers you a list of words from its database that are closest to your search string. Assume that it uses the string alignment algorithm to determine the distance between two strings. These PhD's want to tweak the cost function for the alignment problem to achieve their nefarious goal. The cost function is determined by the gap parameter δ, and the cost of a mismatch between letters x and y, as given by

$$\alpha_{x,y} = \begin{cases} c_0 & \text{if } x = y, \\ c_1 & \text{if } x, y \text{ are both consonants,} \\ c_2 & \text{if } x, y \text{ are both vowels,} \\ c_3 & \text{else.} \end{cases}$$

Originally, $\delta = 3, c_0 = 0, c_1 = c_2 = 1$ amd $c_3 = 2$.

(a) Compute the alignment distance between the strings "google" and "search", using the original cost function.

(b) Suggest a different cost function so that the alignment distance is decreased as much as possible. Please redo part (a) using the new cost function. The rule is that you are only allowed to change the values of $\delta, c_0, c_1, c_2, c_3$ and these values must not be negative.

(c) (Open ended) What are some possible problems with the scheme in (b)?

SOLUTION:

(a) See Table below. Note that the diagonal of this table have cost 0,1,2,3,4,5 and (sic) 7. So the alignment cost is 7.

(b) In order to make the cost as small as possible, we just have to make $c_1 = c_2 = c_3 = 0$. With this cost function, the alignment distance between two string X, Y is simply $\delta \times |(|X| - |Y|)|$. Then the two strings would have an alignment distance of 0. Now, you can be sure that "google" will be among the set of words which will be prompted when you type "search".

(c) The quality of Google Search will degrade: in general, the list of alternatives will be so huge as to be useless. (Those PhD's better clean up their act before their options becomes worthless.)

Table for Question 3, part (a) above:

		g	o	o	g	l	e
	0	3	6	9	12	15	18
s	3	1	4	7	10	13	16
e	6	4	2	5	8	11	14
a	9	7	5	3	6	9	12
r	12	10	8	6	4	7	10
c	15	13	11	9	7	5	8
h	18	16	14	12	10	8	7