

Recurrence: Solve, Prove, Ruminare...

Chris Brown

September 14, 2005

1 The problem

Solve

$$T(0) = 0$$

$$T(n) = \frac{1}{n} \sum_{i=0}^{n-1} T(i) + cn, n > 0.$$

Prove your answer. What is the asymptotic order of $T(n)$? What can you say about generalizations?

2 Solution (creating a guess)

By now in class we've thought about this... did it fit the form for The Master Method? How about just brute force substitution? What's happening? One thing is that ALL the previous $T(k)$'s are involved, which doesn't look like the MM. Substitution looks like a mess. We've sort of seen this kind of thing before... Hmmmm...

OK, it's a telescope job, as it turns out, but you can't do it just as it stands – if you mess around you'll be getting frustrated by that $(1/n)$, but we've seen this problem before in our very first telescope e.g., so try (forgive the notation)

$$nT_n = T_{n-1} + T_{n-2} + \dots + T_1 + T_0 + cn^2.$$

$$(n-1)T_{n-1} = T_{n-2} + T_{n-3} + \dots + T_1 + T_0 + c(n-1)^2.$$

Subtracting (telescoping) makes sense now...

$$nT_n - (n-1)T_{n-1} = T_{n-1} + c(n^2 - (n^2 - 2n + 1)),$$

so

$$n(T_n - T_{n-1}) = c(2n - 1),$$

and

$$(T_n - T_{n-1}) = 2c - \frac{c}{n},$$

thus

$$T_n = T_{n-1} + 2c - \frac{c}{n}.$$

So this is super-simple to add up... we just have n steps of adding the terms involving c and carrying along the sum, so

$$T_n = n2c - c \sum_{i=1}^n \frac{1}{i}.$$

Wow. Out of nowhere has emerged our old friend the harmonic series $H(n)$. As the man says, this happens more in computer science than in the natural sciences...

3 Prove the solution

We know we're in for a proof by mathematical induction. We want to prove the *claim*

$$T_n = 2nc - c \sum_{i=1}^n \frac{1}{i}.$$

We have the definition of the recurrence:

$$T(0) = 0$$

$$T(n) = \frac{1}{n} \sum_{i=0}^{n-1} T(i) + cn.$$

Base case: Does the claim work?

$$T_0 = 0,$$

and furthermore,

$$T_1 = 2c - c \cdot 1 = c,$$

which is easy to check is right by the definition (it equals cn if $n = 1$).

Inductive step:

We assume we know

$$T_0 = 0$$

etc, all the way up to

$$T_k = 2kc - c \sum_{i=1}^k \frac{1}{i},$$

and we want to prove

$$T_{k+1} = 2(k+1)c - c \sum_{i=1}^{k+1} \frac{1}{i}.$$

From the definition we can write (changing notation again)

$$T(k+1) = \frac{1}{k+1}T(k) + \frac{1}{k+1} \sum_{i=0}^{k-1} T(i) + c(k+1).$$

Now remember we're looking for an instance of a solved case of our problem within this mess. That sum looks familiar, and if you refer to the definition you'll see that the definition can be rewritten as

$$\sum_{i=0}^{k-1} T(i) = k(T(k) - ck).$$

So this feels right – we have something fairly neat to substitute in for something sort of messy, and it uses the definition in place of a piece of our problem, so nothing to do but forge on...

$$T(k+1) = \frac{1}{k+1}T(k) + \frac{1}{k+1}(kT(k) - ck^2) + c(k+1).$$

Well this is cool. The terms involving $T(k)$ actually add up to one whole $T(k)$:

$$T(k+1) = T(k) + \frac{1}{k+1}(-ck^2) + c(k+1).$$

Using the claim to re-write the $T(k)$ and doing some algebra on the rest gives

$$T(k+1) = 2kc - c \sum_{i=1}^k \frac{1}{i} + \frac{ck^2 + 2ck + c - ck^2}{k+1}.$$

Things are still a little hazy but we know one thing: we gotta get a $-c/(k+1)$ into that second (harmonic series) term if it's going to be the right sum for the $k+1$ case. So let's create it, subtract it out in the sum and add it back in the rightmost term:

$$T(k+1) = 2kc - c \sum_{i=1}^{k+1} \frac{1}{i} + \frac{ck^2 + 2ck + c + c - ck^2}{k+1}.$$

Simplify that messy last term:

$$T(k+1) = 2kc - c \sum_{i=1}^{k+1} \frac{1}{i} + \frac{2ck + 2c}{k+1},$$

more...

$$T(k+1) = 2kc - c \sum_{i=1}^{k+1} \frac{1}{i} + \frac{2c(k+1)}{k+1},$$

$$T(k+1) = 2kc - c \sum_{i=1}^{k+1} \frac{1}{i} + 2c,$$

et voila...

$$T(k+1) = 2c(k+1) - c \sum_{i=1}^{k+1} \frac{1}{i}.$$

4 Asymptotic complexity

We should remember for the harmonic series,

$$H(n) = \sum_{i=1}^n \frac{1}{i} \approx \ln n$$

with an additive error tending toward a (in fact, Euler's) constant. Thus as $n \rightarrow \infty$, $T(n)$ is in $\Theta(n)$.

5 Afterthought... Constants matter!

Our example recurrence,

$$T(0) = 0$$

$$T(n) = \frac{1}{n} \sum_{i=0}^{n-1} T(i) + cn, n > 0,$$

resembles very much the recurrence describing the expected running time of Quicksort. I'm not sure that analysis is done in detail in CLRS but there's a nice quickie justification for the detailed approach on p. 158. We'll look at the detailed approach in class. Anyway, it's

$$Q(0) = 0$$

$$Q(n) = \frac{2}{n} \sum_{i=0}^{n-1} Q(i) + n, n > 0.$$

That factor of 2 in the recurrence changes the answer from $O(n)$ to $O(n \log n)$. In fact $Q(n) \approx 1.386n \lg n - 1.846n$, as we'll see.

So this raises the obvious question: what if that numerator gets larger? I asked Prof. Ogiwara, who had this to say. [Comments by me in square brackets.]

For $k \geq 3$, $T(n) \in \Theta(n^{k-1})$, although I can't obtain a general solution.

Let's say we have

$$T(n) = (k/n) \sum_{i=0}^{n-1} T(i) + an + b.$$

Here $a + b \geq 0$ so that $T(1) = a + b \geq 0$.

Just to make sure we have a base case with value greater than zero, so the recurrence produces an increasing sum, let m be the first i such that $T(m) > 0$.

Just following the previous approach, we have

$$nT(n) - (n-1)T(n-1) = kT(n-1) + a(2n-1) + b \quad (1)$$

By moving terms we have:

$$nT(n) = (n+k-1)T(n-1) + a(2n-1) + b.$$

This gives an inequality $nT(n) \geq (n+k-1)T(n-1)$. So,

$$\begin{aligned} T(n) &\geq \frac{(n+k-1)(n+k-2)\dots(m+k)}{n(n-1)\dots(m+1)} T(m) \\ &= \frac{(n+k-1)\dots(n+1)}{(m+k-1)\dots(m+1)} T(m). \end{aligned}$$

This is $\Omega(n^{k-1})$. On the other hand, we have an inequality

$$T(n) = \frac{(n+k-1)}{n}T(n-1) + \frac{2a+b}{n} \leq \frac{c(n+k-1)}{n}T(n-1)$$

for some constant c and for all $n \geq m+1$.

By using the same analysis, we have $T(n) = O(c^{k-1}n^{k-1})$. Since c is a constant, we have $T(n) = O(n^{k-1})$.

$k=1$ and $k=2$ are special cases in which Eq. (1) produces much simpler formulas.