Solutions to Math 416 Homework due October 1, Chapter 4

October 4, 2004

1 4.2-3

level	problem size	number of nodes	cost/node	total cost
0	n	1	cn	cn
1	$n/2$	4	$c(n/2)$	$4(cn/2) = 2cn$
2	$n/4$	16	$c(n/4)$	$16(cn/4) = 4cn$
$\vdots$				
j	$n/2^j$	4^j	$cn/2^j$	$cn2^j$
$\vdots$				
$h = \lg(n)$	1	$4^h = n^2$	c	$c4^h = cn^2$

The tree has fanout 4, as indicated in the "number of nodes" column.

The total cost is $cn(1 + 2 + 4 + \cdots + 2^j + \cdots + 2^h) = 2^{h+1}cn$. This is $\Theta(n^2)$. We now prove it by the substitution method. We will show $O(n^2)$ and $\Omega(n^2)$ separately, and introduce a linear and constant term to make the induction go through.

Suppose, for all $n \geq n_0$, we have $T(n) \leq a_2n^2 + a_1n + a_0$, where a_2, a_1, a_0, and n_0 will be specified later, with $a_2 > 0$ and $a_1 \leq 0$. (At this point, it's a guess that $a_1 \leq 0$, but there are only two possibilities. This is "strengthening the inductive hypothesis," as discussed in the text. The constant a_0 is unnecessary, but can't hurt.) First we show the inductive step. We have

$$\begin{aligned}
T(n) &= 4T(\lfloor n/2 \rfloor) + cn \\
&= 4[a_2(\lfloor n/2 \rfloor)^2 + a_1(\lfloor n/2 \rfloor) + a_0] + cn \\
&\leq 4[a_2(n/2)^2 + a_1(n/2 + 1) + a_0] + cn \\
&= a_2n^2 + 2a_1n + 4a_1 + 4a_0 + cn \\
&\leq a_2n^2 + a_1n + a_0,
\end{aligned}$$

provided $2a_1n + 4a_1 + 4a_0 + cn \leq a_1n + a_0$, which is true provided $(a_1 + c)n + 3a_1 + 3a_0 \leq 0$. Asymptotically, we need $a_1 < -c$. Put $a_1 = -2c$, getting $-cn + 3a_0 - 6c \leq 0$, or $cn \geq 3a_0 - 6c$. If $n \geq 1 = n_0$, then we need $7c \geq 3a_0$, or $c \geq (3/7)a_0$. Put $a_0 = c$.

Now we show the base case. We are given that $T(1) \leq c$. This is less than $a_2 \cdot 1^2 + a_1 \cdot 1 + a_0 = a_2 + a_1 + a_0$ provided $c \leq a_2 + a_1 + a_0 = a_2 - 2c + c$; *i.e.*, provided $a_2 \geq 2c$. Put $a_2 = 2c$. We can now confirm: $T(1) = c \leq (2c) \cdot 1^2 - 2c \cdot 1 + c$ and

$$\begin{aligned}
T(n) &= 4T(\lfloor n/2 \rfloor) + cn \\
&= 4[2c(\lfloor n/2 \rfloor)^2 - 2c(\lfloor n/2 \rfloor) + c] + cn \\
&\leq 4[2c(n/2)^2 - 2c(n/2 + 1) + c] + cn \\
&= 2cn^2 - 4cn - 8c + 4c + cn \\
&= 2cn^2 - 3cn - 4c \\
&\leq 2cn^2 - 2cn + c,
\end{aligned}$$

provided $-3cn - 4c \le -2cn + c$, or $cn \ge -3c$, or $n \ge -3$, which is always true.

Similarly, we need to show that, for $n \ge n_1$, we have $T(n) \ge b_2 n^2 + b_1 n + b_0$, for b_2, b_1, b_0, and n_1 to be determined later, with $b_2 > 0$ and $b_1 \le 0$. We have

$$\begin{aligned}
T(n) &= 4T(\lfloor n/2 \rfloor) + cn \\
&= 4[b_2(\lfloor n/2 \rfloor)^2 + b_1(\lfloor n/2 \rfloor) + b_0] + cn \\
&\ge 4[b_2(n/2-1)^2 + b_1(n/2) + b_0] + cn \\
&= 4[b_2(n/2)^2 - b_2 n + b_2 + b_1(n/2) + b_0] + cn \\
&= b_2 n^2 + (2b_1 - 4b_2 + c)n + 4(b_2 + b_0) \\
&\ge b_2 n^2 + b_1 n + b_0,
\end{aligned}$$

provided $(2b_1 - 4b_2 + c)n + 4(b_2 + b_0) \ge b_1 n + b_0$, or $(b_1 - 4b_2 + c)n + 4b_2 + 3b_0 \ge 0$. Put $b_2 = 1$; we need $(b_1 - 4 + c)n + 4 + 3b_0 \ge 0$. Asymptotically, we need $b_1 > 4 - c$; put $b_1 = 5 - c$, which will give what we need for large enough n. Next, consider the base case. We have $T(1) = c$ which is at least $b_2 \cdot 1^2 + b_1 \cdot 1 + b_0 = 1 + (5 - c) + b_0$ provided $c \ge 6 - c + b_0$, or $b_0 \le 2c - 6$. Put $b_0 = 2c - 6$.

We now verify that $T(n) \ge n^2 + (5-c)n + 2c - 6$: $T(1) = c \ge 1 \cdot 1^2 + (5-c) \cdot 1 + (2c-6)$. Also, assuming $c \ge 5$, we get

$$\begin{aligned}
T(n) &= 4T(\lfloor n/2 \rfloor) + cn \\
&= 4[(\lfloor n/2 \rfloor)^2(5-c)(\lfloor n/2 \rfloor) + (2c-6)] + cn \\
&\ge 4(n/2-1)^2 + (5-c)(n/2) + (2c-6)] + cn \\
&= 4[(n/2)^2 - n + 1 + (5-c)(n/2) + (2c-6)] + cn \\
&= n^2 + (2(5-c) - 4 + c)n + 4(1 + (2c-6)) \\
&= n^2 + (6-c)n - 20 + 12c \\
&\ge n^2 + (5-c)n + (2c-6),
\end{aligned}$$

provided $n - 20 + 12c \ge 2c - 6$, or $n \ge -10c + 14 \ge -36$, since $c \ge 5$. This is always true.

For this problem, it would be ok to show the result only for n a power of 2. (Formally, one can tell from the form of the recurrence that $T(n)$ is increasing and that the solution is some polynomial in n, say of degree d. It follows that we can round n up or down to some m a power of 2 and affect the solution by at most the factor 2^d.) Often the algorithmics is easier than the calculus: often, at top level, we can reduce our problem of size n to the problem of size m with no additional overhead.

2 Problem 4-1

Note: In some cases, we can get an exact solution in terms of $T(1)$. Here we often settle for constant-factor Θ notation, which is worth full points.

a. $T(n) = 2T(n/2) + n^3$: By master method, $T(n) = \Theta(n^3)$.

   To get an exact solution, put $T(n) = a_3 n^3 + a_2 n^2 + a_1 n + a_0$, prove that this holds by induction for all n at least some n_0, and see what values of a_3, a_2, a_1, a_0, and n_0 fall out, as above. Or, $T(n) = 2T(n/2) + n^3 = 2(2T(n/4) + (n/2)^3) + n^3 = \cdots$. This is $n^3(1 + 1/4 + (1/4)^2 + ...)$, which is close to $(4/3)n^3$. There is also a contribution of c for each of n leaves, totaling cn, which is not dominant, provided n is large enough, compared with c.

b. $T(n) = T(9n/10) + n$: By master method, $T(n) = \Theta(n)$. Note: here we need to take the convention that this means $T(n) = T(\lfloor 9n/10 \rfloor) + n$ or $T(n) = T(\lceil 9n/10 \rceil) + n$. To get intuition, try a few values: $T(n) = T(9n/10) + n = (T(81n/100) + 9n/10) + n = ((T((9/10)^3 n) + (9/10)^2 n) + 9n/10) + n....$ Ultimately, $T(n) = T(1) + n(1 + (9/10) + (9/10)^2 + \cdots + (9/10)^{\log_{10/9}(n)})$. The finite series is close to the infinite series, $T(n) = T(1) + 10n$.

c. $T(n) = 16T(n/4) + n^2$: By the master method, $T(n) = \Theta(n^2 \lg(n))$.

d. $T(n) = 7T(n/3) + n^2$: By the master method, $T(n) = \Theta(n^2)$, since $\log_3(7) < \log_3(9) = 2$. Note: strict inequality, so, for some $\epsilon > 0$ independent of n, we have $\log_3(7) < 2 - \epsilon$.

e. $T(n) = 7T(n/2) + n^2$: By the master method, $T(n) = n^{\log_2(7)}$, since $\log_2(7) > \log_2(4) = 2$.

f. $T(n) = 2T(n/4) + \sqrt{n}$: By the master method, $T(n) = \Theta(\sqrt{n}\log(n))$.

g. $T(n) = T(n-1) + n$: This is $\Delta T(n) = n + 1$. Note that $\sum_{0 \le k < n} \Delta T(k) = T(n) - T(0)$ by telescoping: $(T(k) - T(k-1)) + (T(k-1) - T(k-2)) + \cdots + (T(1) - T(0))$. Thus

$$\begin{aligned} T(n) - T(0) &= \sum_{0 \le k < n} (k+1) \\ &= \binom{k}{2} + \binom{k}{1}\Big|_0^n \\ &= \frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}. \end{aligned}$$

Alternatively, get intuition by starting at 0: $T(1) = T(0) + 1$, $T(2) = T(1) + 2 = T(0) + 1 + 2$, $T(3) = T(2) + 3 = T(0) + 1 + 2 + 3$, etc.

h. $T(n) = T(\sqrt{m}) + 1$: Here we'll assume n is of the form 2^{2^m}. We get $T(2^{2^m}) = T(2^{2^{m-1}}) + 1 = T(2^{2^{m-2}}) + 2 = \cdots$. The height of the recursion tree is $m = \lg\lg(n)$, so we get $T(n) = \lg\lg(n) + T(0)$. Note: if n is not of this form, if $T(n)$ describes a runtime, and if, in the application, we can reduce our problem to a larger problem, we can round n up to $2^{2^{\lceil \lg\lg(n) \rceil}} \le 2^{2^{\lg\lg(n)+1}} = 2^{2 \cdot 2^{\lg\lg(n)}} = \left(2^{2^{\lg\lg(n)}}\right)^2 = n^2$. Then $\lg\lg(n^2) \le \lg(\lg(n) + 1) = \lg\lg(n)(1 + o(1))$. Thus the asymptotics are the same for all n. Alternatively, put $n = 2^m$ and $S(m) = T(2^m)$. Then $S(m) = T(n) = T(\sqrt{n}) + 1 = S(m/2) + 1$. By the master method, $S(m) = \Theta(\lg(m))$. Thus $T(n) = S(m) = \Theta(\lg\lg(n))$.