Solving Recurrence Relations

Some Examples

1. Give an asymptotic upper bound for the recurrence $T(n) = 7T(n/2) + 15n^2/4$.
   Solution: We try the *Master Theorem*. We have $a = 7$, $b = 2$, and $d = 2$. Since $7 > 2^2$, the third case applies so $T(n) = \Theta(n^{\log_2 7})$, which is about $\Theta(n^{2.8})$

2. Give a tight bound for the recurrence $T(n) = T(\sqrt{n}) + 1$, where $T(2) = 1$
   Solution: We can see that

$$\begin{aligned} T(n) &= T(n^{1/2}) + 1 \\ &= T(n^{1/4}) + 1 + 1 \\ &= T(n^{1/8}) + 1 + 1 + 1 \\ &= T(n^{1/2^k}) + k \end{aligned}$$

   If we can determine when $n^{1/2^k} = 2$, we can obtain a solution. Taking logs (base 2) on both sides, we get

$$\log_2(n^{1/2^k}) = \log_2 2.$$

   We apply the power-inside-a-log rule and the fact that $\log_2 2 = 1$ to get

$$(1/2^k)\log_2 n = 1.$$

   Multiplying both sides by 2^k and flipping it around, we get

$$2^k = \log_2 n.$$

   Again taking logs, we get

$$k = \log_2 \log_2 n.$$

   Therefore,

$$\begin{aligned} T(n) &= T(n^{1/2^{\log_2 \log_2 n}}) + \log_2 \log_2 n \\ &= T(2) + \log_2 \log_2 n \\ &= 1 + \log_2 \log_2 n \\ &= \Theta(\log_2 \log_2 n). \end{aligned}$$

3. Solve the recurrence relation $T(n) = 2T(n/2) + n^3$, $T(1) = 1$.
Solution:
We start by backward substitution:

$$\begin{aligned} T(n) &= 2T(n/2) + n^3 \\ &= 2[2T(n/4) + (n/2)^3] + n^3 \\ &= 2[2T(n/4) + n^3/8)] + n^3 \\ &= 2^2T(n/4) + n^3/4 + n^3 \end{aligned}$$

Notice that in the second line we have $(n/2)^3$ and not n^3 or $n^3/2$. This may be more clear if rewrite the formula using k: $T(k) = 2T(k/2) + k^3$. When applying the formula to $T(n/2)$, we have $k = n/2$, so we get

$$T(n/2) = 2T((n/2)/2) + (n/2)^3 = 2T(n/2) + n^3/8.$$

Continuing,

$$\begin{aligned} T(n) &= \ldots \\ &= 2^2T(n/4) + n^3/4 + n^3 \\ &= 2^2[2T(n/8) + (n/4)^3] + n^3/4 + n^3 \\ &= 2^2[2T(n/8) + n^3/4^3] + n^3/4 + n^3 \\ &= 2^3T(n/8) + n/4^2 + n^3/4 + n^3. \end{aligned}$$

Notice that I use 2 or more steps for every iteration–I do one substitution and then simplify it before moving on to the next substitution. This helps to ensure I don't make algebra mistakes and that I can write it out in a way that helps me see a pattern.

Also notice that we can write the last line as

$$2^3T(n/2^3) + n/4^2 + n^3/4^1 + n^3/4^0,$$

so it appears that we can generalize this to

$$2^kT(n/2^k) + \sum_{i=0}^{k-1} n^3/4^i.$$

The sum starts at $i = 0$ (not 1) and goes to $k-1$ (not k). It is easy to get either (or both) of these wrong if you aren't careful. We should be careful to make sure we have seen the correct pattern.[1] Continuing (with a few more steps shown to make all of the algebra as clear as possible), we get

$$\begin{aligned} T(n) &= \ldots \\ &= 2^3T(n/2^3) + n/4^2 + n^3/4^1 + n^3/4^0 \\ &= 2^3[2T(n/2^4) + (n/2^3)^3] + n/4^2 + n^3/4^1 + n^3/4^0 \\ &= 2^3[2T(n/2^4) + n^3/2^9] + n/4^2 + n^3/4^1 + n^3/4^0 \\ &= 2^4T(n/2^4) + n^3/2^6 + n/4^2 + n^3/4^1 + n^3/4^0 \\ &= 2^4T(n/2^4) + n^3/4^3 + n/4^2 + n^3/4^1 + n^3/4^0 \\ &= \ldots \\ &= 2^kT(n/2^k) + \sum_{i=0}^{k-1} n^3/4^i. \end{aligned}$$

Notice that this *does* seem to match the pattern we saw above. We can evaluate the sum to simplify it a little more:

[1]Too often I have seen students make a pattern out of 2 iterations. Not only is this not enough iterations to be sure of anything, the pattern they usually come up with only holds for the last iteration they did. The pattern has to match every iteration. To be safe, go one more iteration after you identify the pattern to verify that it is correct.

$$
\begin{aligned}
T(n) &= \ldots \\
&= 2^k T(n/2^k) + \sum_{i=0}^{k-1} n^3/4^i \\
&= 2^k T(n/2^k) + n^3 \sum_{i=0}^{k-1} 1/4^i \\
&= 2^k T(n/2^k) + n^3 \sum_{i=0}^{k-1} (1/4)^i \\
&= 2^k T(n/2^k) + n^3 \left(\frac{1-(1/4)^k}{1-1/4}\right) \\
&= 2^k T(n/2^k) + n^3 (4/3)(1-(1/4)^k)
\end{aligned}
$$

We are almost done. We just need to find a k that allows us to get rid of the recursion. Thus, we need to determine what value of k makes $T(n/2^k) = T(1) = 1$. In other words, we need k such that

$$n/2^k = 1.$$

This is equivalent to

$$n = 2^k.$$

Taking log (base 2) of both sides, we obtain

$$\log_2 n = \log_2(2^k) = k \log_2 2 = k.$$

So $k = \log_2 n$. We plug in k and use the fact that $2^{\log_2 n} = n$ along with the exponent rules to obtain

$$
\begin{aligned}
T(n) &= \ldots \\
&= 2^k T(n/2^k) + n^3(4/3)(1-(1/4)^k) \\
&= 2^{\log_2 n} T(n/2^{\log_2 n}) + n^3(4/3)(1-(1/4)^{\log_2 n}) \\
&= nT(1) + n^3(4/3)\left(1 - \frac{1}{(2^2)^{\log_2 n}}\right) \\
&= n \cdot 1 + n^3(4/3)\left(1 - \frac{1}{(2^{\log_2 n})^2}\right) \\
&= n + n^3(4/3)\left(1 - \frac{1}{n^2}\right) \\
&= n + \frac{4}{3}n^3 - \frac{4}{3}n \\
&= \frac{4}{3}n^3 - \frac{1}{3}n.
\end{aligned}
$$

So we have that $T(n) = \frac{4}{3}n^3 - \frac{1}{3}n = \Theta(n^3)$. Notice that we can get the Θ-bound much easier with the Master Method. We have $a = 2$, $b = 2$, and $d = 3$, and since $2 < 2^3$, $T(n) = \Theta(n^3)$, which luckily matches what we obtained above. Of course by doing it the hard way, we actually have the exact answer instead of just a bound.