csce750 — Analysis of Algorithms
Fall 2019 — Lecture Notes: Solving Recurrences — Substitution Method

This document contains slides from the lecture, formatted to be suitable for printing or individual reading, and with some supplemental explanations added. It is intended as a supplement to, rather than a replacement for, the lectures themselves — you should not expect the notes to be self-contained or complete on their own.

1 *Definition*

A **recurrence** is an equation or inequality that describes a function in terms of its own value on smaller inputs. CLRS 4.3–4.5

We've already seen one example (for the run time of MergeSort):

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq 1 \\ 2T(\frac{n}{2}) + \Theta(n) & \text{otherwise} \end{cases}$$

Recurrences are important because they are the primary tool for analyzing recursive algorithms.

We'll look at three different ways to solve recurrences.

- Substitution method
- Recursion trees
- The Master theorem

2 *Smoothness rule*

Definition A non-negative function $f(n)$ is called **smooth** if

$$f(2n) \in \Theta(f(n)).$$

Example: $f(n) = n^3$ is a smooth function, because

$$f(2n) = (2n)^3 = 8n^3 = \Theta(n^3).$$

Example: $g(n) = 2^n$ is not a smooth function, because

$$g(2n) = 2^{2n} \neq \Theta(2^n).$$

Smoothness rule (informally): Suppose we show, for some $b \geq 2$, that $f(n) = \Theta(g(n))$ when n is a power of b. Then, if $f(n)$ is a smooth function, we have $f(n) = \Theta(g(n))$ for *all* n.

Smoothness rule (even more informally): Most of the time, floors and ceilings do not affect the asymptotic growth rate.

3 *Substitution method*

The **substitution method** for solving recurrences has two parts.

1. **Guess** the correct answer.
2. **Prove** by induction that your guess is correct.

4 *Example*

Use the substitution method to solve $T(n) = 2T(n/2) + n$.

Guess: $T(n) = O(n \lg n)$

Proof: Use induction on n to show that there exist c and n_0 for which $T(n) \le cn \lg n$ for all $n \ge n_0$.

- Base case: Almost always omitted, because $T(n) = \Theta(1)$ when n is sufficiently small, so we can always choose c large enough.

  Details: CLRS 84

- Induction step: Assume that $T(m) \le cm \lg m$ for all $m < n$, to prove that $T(n) \le cn \lg n$.

  (We'll find restrictions on c and n_0 along the way.)

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\le 2 \cdot c(n/2) \lg(n/2) + n \\ &= cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\le cn \lg n \qquad [\text{ when } c \ge 1] \end{aligned}$$

In the last step, we replace $-cn + n$ with 0. This increases the sum if

$$-cn + n \le 0.$$

Solve for c to get the constraint $c \ge 1$.

5 *Be careful!*

Substitution proofs must ensure that they use the **same constant** as in the inductive hypothesis.

Here's an example of how to "prove" (incorrectly!) that $T(n) = O(n)$.

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\le 2c(n/2) + n \\ &= (c+1)n \\ &= O(n) \end{aligned}$$

> *The problem is that we have not proved the* exact form of the inductive hypothesis. *In particular, the constant c we use when we substitute at the beginning must be the same c we have at the end of the inequalities.*

6 Even correct guesses can lead to dead ends

Show that $T(n) = 5T(n/2) + n^2$ is $O(n^{\log_2 5})$.

Attempt 1: Use induction to (try to) show that, for some c,

$$T(n) \leq cn^{\log_2 5}.$$

"Proof:"

$$\begin{aligned} T(n) &= 5T(n/2) + n^2 \\ &\leq 5\left(c\left(\frac{n}{2}\right)^{\log_2 5}\right) + n^2 \\ &= 5\left(c\frac{n^{\log_2 5}}{2^{\log_2 5}}\right) + n^2 \\ &= cn^{\log_2 5} + n^2 \end{aligned}$$

A dead end! No choice of c makes this inequality true.

7 Proving a stronger bound

Show that $T(n) = 5T(n/2) + n^2$ is $O(n^{\log_2 5})$.

Attempt 2: Use induction to show that, for some c and some a,

$$T(n) \leq cn^{\log_2 5} - an^2.$$

> *Note that if we can show that $T(n) \leq cn^{\log_2 5} - an^2$ for some positive constant a, then we know immediately that $T(n) \leq cn^{\log_2 5}$, which is sufficient to show that $T(n) \in O(n^{\log_2 5})$. So this strong bound really is doing the job that we need it to.*
>
> *The choice of adding $-an^2$ here is based on the dead end from the previous slide — we had an extra n^2 term, and we're hoping that the new term in the inductive hypothesis will counteract that. This doesn't always work out, but it often does, and it's a good place to start.*

Proof:

$$
\begin{aligned}
T(n) &= 5T(n/2) + n^2 \\
&\leq 5\left(c\left(\frac{n}{2}\right)^{\log_2 5} - a\left(\frac{n}{2}\right)^2\right) + n^2 \\
&= 5\left(c\frac{n^{\log_2 5}}{2^{\log_2 5}} - a\frac{n^2}{4}\right) + n^2 \\
&= cn^{\log_2 5} + \left(1 - \frac{5a}{4}\right)n^2 \\
&\leq cn^{\log_2 5} - an^2 \qquad [a \geq 4]
\end{aligned}
$$

This is enough to conclude $T(n) = O(n^{\log_2 5})$, because $cn^{\log_2 5} - 4n^2 \leq cn^{\log_2 5}$.

In the last step of the proof, we replace $\left(1 - \frac{5a}{4}\right)$ *with* $-a$*. We want* $-a$*, because we need to match the exact form of the inductive hypothesis.*

This replacement increases the expression (or leaves it unchanged), making the $\leq$ *we write there correct, when*

$$\left(1 - \frac{5a}{4}\right) \leq -a.$$

Solving for a *(first multiply both sides by* 4*, then add* $5a$ *to both sides) we get:*

$$
\begin{aligned}
4 - 5a &\leq -4a \\
4 &\leq a
\end{aligned}
$$

Hence the constraint $a \geq 4$*.*

8 Change of variables

Solve the recurrence $T(n) = 2T(\sqrt{n}) + \lg n$.

Solution: Change of variables. Let $m = \lg n$ and $S(m) = T(2^m)$.

Note that $n = 2^m$ and $\sqrt{n} = 2^{m/2}$.

Then we get:

$$
\begin{aligned}
T(n) &= T(2^m) \\
&= 2T(2^{m/2}) + \lg(2^m) \\
&= 2S(m/2) + m \\
&= O(m \log m) \\
&= O(\log n \log \log n)
\end{aligned}
$$

Recall that we've solved the recurrence $S(m) = 2S(m/2) + n$ already, a few slides back.