CIS 575. Introduction to Algorithm Analysis
Material for February 7, 2024

The Master Theorem: the Intuition

The topic of this note is mentioned in *Cormen*'s Section 4.4.

1 Solving Recurrences by Unfolding

We have seen that the substitution method enables us to **verify** a **proposed** solution to a given recurrence. In the next notes, we shall present a recipe for **finding** the solution to a given recurrence.

To motivate that recipe, let us look at the recurrence for MergeSort, slightly modified and rearranged into

$$T(n) = n + 2T(\frac{n}{2})$$

Recall that we used the substitution method to verify that $T(n) \in O(n \lg(n))$. But there is a simpler, though (for now) less rigorous, way to show $T(n) \in \Theta(n \lg(n))$: repeatedly *unfold* the recurrence into

$$\begin{aligned}
T(n) &= n + 2T(\frac{n}{2}) \\
&= n + 2(\frac{n}{2} + 2T(\frac{n}{4})) \\
&= n + 2(\frac{n}{2} + 2(\frac{n}{4} + 2T(\frac{n}{8}))) \\
&\approx \overbrace{n + n + n + \ldots + n}^{\lg n \text{ times}} + n \cdot T(1) \\
&\approx n \lg(n)
\end{aligned}$$

For a *more general* recurrence of the form (with q a non-negative real)

$$T(n) = n^q + aT(\frac{n}{b})$$

a repeated unfolding will result in

$$\begin{aligned}
T(n) &= n^q + aT(\frac{n}{b}) \\
&= n^q + a(\frac{n^q}{b^q} + aT(\frac{n}{b^2})) \\
&= n^q + a(\frac{n^q}{b^q} + a(\frac{n^q}{b^{2q}} + aT(\frac{n}{b^3})))
\end{aligned}$$

$$
\begin{aligned}
&\approx\; n^q + \frac{a}{b^q}n^q + (\frac{a}{b^q})^2 n^q + \ldots + (\frac{a}{b^q})^{\log_b(n)} n^q + a^{\log_b(n)} T(1) \\
&\approx\; (1 + \frac{a}{b^q} + (\frac{a}{b^q})^2 + \ldots (\frac{a}{b^q})^{\log_b(n)}) n^q
\end{aligned}
$$

With $c = \dfrac{a}{b^q}$ we thus have

$$T(n) \approx (1 + c + c^2 + \ldots + c^{\log_b(n)}) n^q$$

where the sum can be of 3 different kinds:

1. if $c < 1$, that is if $\mathbf{b^q > a}$, then the terms get smaller and smaller, and in fact

   $$1 + c + c^2 + \ldots \leq \frac{1}{1-c}$$

   which suggests that in this case we have $\mathbf{T(n) \in \Theta(n^q)}$.

2. if $c = 1$, that is if $\mathbf{b^q = a}$, then all the $\log_b(n)$ terms contribute equally, and thus $\mathbf{T(n) \in \Theta(n^q\ \log_b(n))}$.

3. if $c > 1$, that is if $\mathbf{b^q < a}$, then the terms grow bigger and bigger and what "matters" is (modulo a constant factor) the last term which is given by

   $$c^{\log_b(n)} n^q = \frac{a^{\log_b(n)}}{(b^q)^{\log_b(n)}} n^q = \frac{a^{\log_b(n)}}{(b^{\log_b(n)})^q} n^q = \frac{a^{\log_b(n)}}{n^q} n^q = a^{\log_b(n)} = n^{\log_b(a)}$$

   which suggests that in this case we have $\mathbf{T(n) \in \Theta(n^{\log_b(a)})}$.

In the next note, we shall summarize our findings as a general and widely applicable result.