Data Structures and Algorithms
Lecture Outline
February 04, 2016

Simplified Master Theorem. Let $a \geq 1$, $b > 1$ be constants and let $T(n)$ be the recurrence

$$T(n) = aT\left(\frac{n}{b}\right) + \Theta(n^k)$$

defined for $n \geq 0$ (we assume that n is a power of b, though this does not make a difference in asymptotic analysis). The base case, $T(1)$ can be any constant value. Then
Case 1: if $a > b^k$, then $T(n) \in \Theta(n^{\log_b a})$.
Case 2: if $a = b^k$, then $T(n) \in \Theta(n^k \log_b n)$.
Case 3: if $a < b^k$, then $T(n) \in \Theta(n^k)$.

Example. Using Master Theorem, prove bounds on $T(n)$ for the following recurrences.

a. $T(n) = T(n/2) + n^2/2 + n$

b. $T(n) = 2T(n/4) + \sqrt{n}$

c. $T(n) = 3T(n/2) + (3/4)n$

d. $T(n) = 2T(n/4) + 5$

Solution.

a. $a = 1, b = 2$, and $k = 2$, thus case 3 applies and we get $T(n) = O(n^2)$.

b. $a = 2, b = 4$, and $k = 1/2$, thus case 2 applies and we get $T(n) = O(\sqrt{n} \log n)$.

c. $a = 3, b = 2$, and $k = 1$, thus case 1 applies and we get $T(n) = O(n^{\log_2 3})$.

d. $a = 2, b = 4$, and $k = 0$, thus case 1 applies and we get $T(n) = O(n^{\log_4 2}) = O(\sqrt{n})$.

There are recurrences for which none of the above cases apply. In that case, we solve them using other techniques that we have learned. An example of such a recurrence is $T(n) = T(n/4) + \lg n$.

Example. Consider the following recurrence. We assume n is a power of 2.

$$T(n) = \begin{cases} 2T(n/2) + n \lg n, & n \geq 2 \\ 1, & \text{otherwise} \end{cases}$$

Express $T(n)$ as $\Theta(f(n))$.

Solution. Note that the above recurrence cannot be solved using the Simplified Master Theorem. We use the method of expansion.

$$
\begin{aligned}
T(n) &= 2T(n/2) + n\lg n \\
&= 2^2T(n/2^2) + n\lg(n/2) + n\lg n \\
&= 2^3T(n/2^3) + n\lg(n/4) + n\lg(n/2) + n\lg n \\
&\ldots \\
&\ldots \\
&= 2^kT(n/2^k) + n\lg(n/2^{k-1}) + \ldots + n\lg(n/4) + n\lg(n/2) + n\lg n \\
&= 2^kT(n/2^k) + n\sum_{i=0}^{k-1}\lg(n/2^i) \\
&= 2^kT(n/2^k) + n\sum_{i=0}^{k-1}(\lg n - \lg 2^i) \\
&= 2^kT(n/2^k) + n\sum_{i=0}^{k-1}(\lg n - i) \\
&= 2^kT(n/2^k) + kn\lg n - nk(k-1)/2 \\
&= 2^kT(n/2^k) + kn\lg n - nk^2/2 + nk/2
\end{aligned}
$$

The recursion bottoms out when $n/2^k = 1$, i.e., $k = \lg n$. Thus, we get

$$
\begin{aligned}
T(n) &= nT(1) + n\lg^2 n - n\lg^2 n/2 + n\lg n/2 \\
&= n + n\lg^2 n/2 + n\lg n/2 \\
&= \Theta(n\lg^2 n)
\end{aligned}
$$