"Master Theorem" – Revised Version

Let

$$T(n) = \begin{cases} aT(n/b) + cn^d & n > 1 \\ f & n = 1 \end{cases}$$

where $n = b^k$.

Then

$$T(n) = \begin{cases} \left(f + \frac{c}{\frac{a}{b^d}-1}\right) n^{\log_b a} + \frac{cn^d}{1-\frac{a}{b^d}} = \begin{cases} \Theta(n^{\log_b a}) & a > b^d \\ \Theta(n^d) & a < b^d \end{cases} \\ n^d(f + c\log_b n) = \Theta(n^d \log_b n) & a = b^d \end{cases}.$$

Proof:

$$\begin{aligned}
T(n) &= aT(\frac{n}{b}) + cn^d \\
&= a\left[aT(\frac{n}{b^2}) + c(\frac{n}{b})^d\right] + cn^d \\
&= a\left[a\left[aT(\frac{n}{b^3}) + c(\frac{n}{b^2})\right] + c(\frac{n}{b})^d)\right] + cn^d \\
&= a^3T(\frac{n}{b^3}) + a^2c(\frac{n}{b^2})^d + ac(\frac{n}{b})^d + cn^d \\
&\quad\vdots \\
&= a^kT(\frac{n}{b^k}) + a^{k-1}c(\frac{n}{b^{k-1}})^d + a^{k-2}c(\frac{n}{b^{k-2}})^d + \cdots + cn^d \\
&= a^kT(\frac{n}{b^k}) + \sum_{j=0}^{k-1} a^j c(\frac{n}{b^j})^d \\
&= a^kT(\frac{n}{b^k}) + cn^d \sum_{j=0}^{k-1} (\frac{a}{b^d})^j \\
&= a^kT(\frac{n}{b^k}) + cn^d \frac{1-(\frac{a}{b^d})^k}{1-\frac{a}{b^d}} \qquad \text{for } \tfrac{a}{b^d} \neq 1 \\
&= a^kT(\frac{n}{b^k}) + cn^d \frac{1-\frac{a^k}{(b^k)^d}}{1-\frac{a}{b^d}} \\
&= a^{\log_b n}T(\frac{n}{n}) + cn^d \frac{1-\frac{a^{\log_b n}}{n^d}}{1-\frac{a}{b^d}} \qquad \text{substituting } n = b^k \text{ or } k = \log_b n \\
&= fn^{\log_b a} + c\frac{n^d - n^{\log_b a}}{1-\frac{a}{b^d}}
\end{aligned}$$

If $\frac{a}{b^d} = 1$ then

$$\begin{aligned}
T(n) &= a^k T(\frac{n}{b^k}) + cn^d \sum_{j=0}^{k-1} (\frac{a}{b^d})^j \\
&= a^k T(\frac{n}{b^k}) + cn^d k \\
&= a^{\log_b n} T(\frac{n}{n}) + cn^d \log_b n \qquad \text{substituting } n = b^k \text{ or } k = \log_b n \\
&= f n^{\log_b a} + cn^d \log_b n
\end{aligned}$$

$\square$

Intuition: The solution of this recurrence can be thought of in terms of a tree, where the root contributes cn^d and has a children. Each child contributes $c(\frac{n}{b})^d$ and has a children. Each grandchild contributes $c(\frac{n}{b^2})^d$ and has a children. Etc. The tree has height $\log_b n$, and there are $n^{\log_b a}$ leaves.

- When the branching factor is high ($a > b^d$), a major portion of the contribution occurs at the leaves. This yields $\Theta(n^{\log_b a})$.
- When the branching factor is low ($a < b^d$), a major portion of the contribution occurs at the root. This yields $\Theta(n^d)$.
- When the branching factor is "in between" ($a = b^d$), each nonleaf level of the tree contributes the same amount. There are $\log_b n$ nonleaf levels each of which contributes cn^d. The n^d leaves each contribute f. The total is $n^d(f + c\log_b n)$.

Summing solutions: If

$$T(n) = \begin{cases} aT(n/b) + \sum c_i n^{d_i} & n > 1 \\ f & n = 1 \end{cases}$$

then we can just sum the solutions of each recurrence:

$$T_i(n) = \begin{cases} aT_i(n/b) + c_i n^{d_i} & n > 1 \\ 0 & n = 1 \end{cases}$$

and add in $fn^{\log_b a}$ for the contribution from the leaves. For example, MergeSort has the following recurrence:

$$T(n) = \begin{cases} 2T(n/2) + n - 1 & n > 1 \\ 0 & n = 1 \end{cases}$$

The $+n$ term gives a solution of $n \lg n$, the -1 term gives a solution of $-n+1$, and the leaves contribute $0 \cdot n^{\log_2 2}$, so the full solution is

$$T(n) = n \lg n - n + 1$$