Master Method

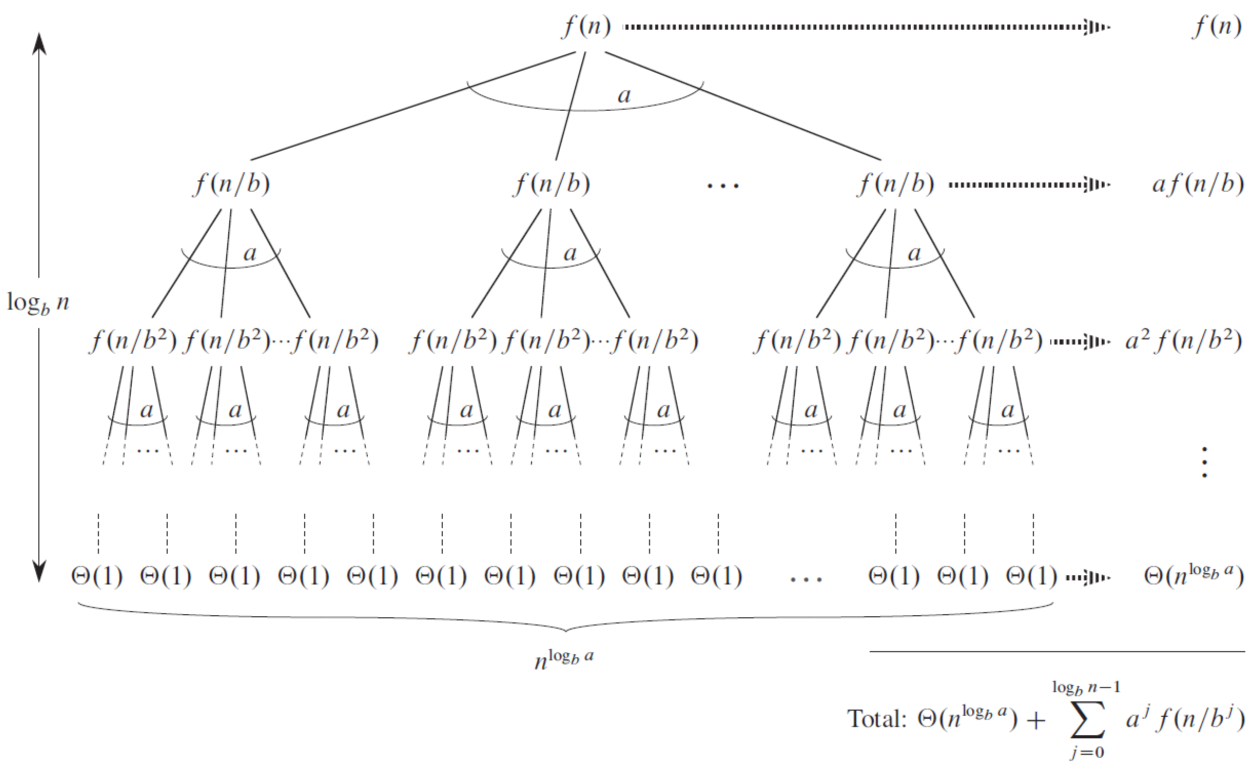
The **master method** provides a "cookbook" method for solving recurrences of the form

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

where $a \geq \mathbf{1}$ and $b > \mathbf{1}$ are **constants** and $f(n)$ is an asymptotically **positive** function.

Note: Here, we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$.

The recursion tree for the recurrence $T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$ is shown in the following figure (once again, for the sake of simplicity, we assume that n is an exact power of b):



The tree is a **complete** a-ary tree with $n^{\log_b a}$ leaves and height $\log_b n$. The cost of the nodes at each depth is shown at the right, and their sum is given in the following equation:

$$T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n - 1} a^j\, f\left(\frac{n}{b^j}\right)$$

Let $g(n)$ be a function defined (over exact powers of b) as follows:

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j\, f\left(\frac{n}{b^j}\right)$$

then $g(n)$ has the following asymptotic bounds:

Case 1:

If $\boldsymbol{f(n) = O\left(n^{\log_b a-\epsilon}\right)}$ for some constant $\epsilon > 0$, then $g(n) = O\left(n^{\log_b a}\right)$. Hence,

$$\begin{aligned} T(n) &= \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n-1} a^j f\left(\frac{n}{b^j}\right) \\ &= \Theta\left(n^{\log_b a}\right) + O\left(n^{\log_b a}\right) \\ &\boldsymbol{= \Theta\left(n^{\log_b a}\right)} \end{aligned}$$

Case 2:

If $\boldsymbol{f(n) = \Theta\left(n^{\log_b a}\right)}$, then $g(n) = \Theta\left(n^{\log_b a} \lg n\right)$. Hence,

$$\begin{aligned} T(n) &= \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n-1} a^j f\left(\frac{n}{b^j}\right) \\ &= \Theta\left(n^{\log_b a}\right) + \Theta\left(n^{\log_b a} \lg n\right) \\ &\boldsymbol{= \Theta\left(n^{\log_b a} \lg n\right)} \end{aligned}$$

Case 3:

If $af(n/b) \leq cf(n)$ for some constant $c < 1$ and for all sufficiently large n, then $g(n) = \Theta(f(n))$.

$$\begin{aligned} T(n) &= \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n-1} a^j f\left(\frac{n}{b^j}\right) \\ &= \Theta\left(n^{\log_b a}\right) + \Theta(f(n)) \end{aligned}$$

Moreover, we also have that if $af(n/b) \leq cf(n)$ then $\boldsymbol{f(n) = \Omega\left(n^{\log_b a+\epsilon}\right)}$ for some constant $\epsilon > 0$. Hence,

$$T(n) = \Theta\left(n^{\log_b a}\right) + \Theta\left(f(n)\right) = \boldsymbol{\Theta\left(f(n)\right)}$$

Master Method

For each of the following recurrences, determine whether or not the Master Method can be used to solve the recurrence. In affirmative scenarios, solve the recurrence using Master Method (basic version only).

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

b) $T(n) = T(n - 2) + n^2$

c) $T(n) = 10T(n/2) + \Theta(1)$

d) $T(n) = 2T(n - 1) + \Theta(1)$

e) $T(n) = T(n - 1) + 1/n$

f) $T(n) = 4T(n/2) + n^2\sqrt{n}$

g) $T(n) = 2T(n/2) + n/\lg n$

h) $T(n) = \sqrt{n}\, T(n/2) + n \lg n$

i) $T(n) = 0.5\, T(n/2) + n^2$

j) $T(n) = 5T(n/25) - n^5$

k) $T(n) = 2T(n/4) + 1$

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

m) $T(n) = 2T(n/4) + n$

n) $T(n) = 2T(n/4) + n^2$