Why should the Master Theorem be true?

Consider $T(n) = aT(n/b) + f(n)$.

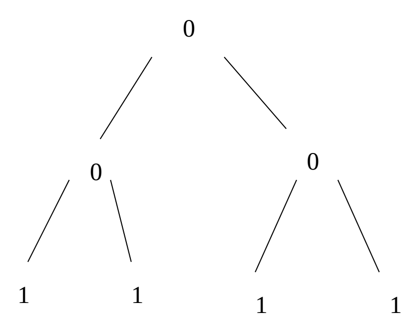
Suppose $f(n)$ is small enough

Say $f(n) = 0$, ie. $T(n) = aT(n/b)$.

Then we have a recursion tree where the only contribution is at the leaves.

There will be $\log_b n$ levels, with a^l leaves at level l.

$$T(n) = a^{\log_b n} = n^{\log_b a} \quad \text{Theorem 2.9 in CLR}$$



so long as $f(n)$ is small enough that it is dwarfed by this, we have case 1 of the Master Theorem!

Suppose f(n) is large enough

If we draw the recursion tree for $T(n) = aT(n/b) + f(n)$.

If $f(n)$ is a big enough function, the one top call can be bigger than the sum of all the little calls.

Example: $f(n) = n^3 > (n/3)^3 + (n/3)^3 + (n/3)^3$. In fact this holds unless $a \geq 27$!

In case 3 of the Master Theorem, the additive term dominates.

In case 2, both parts contribute equally, which is why the log pops up. It is (usually) what we want to have happen in a divide and conquer algorithm.

See if you can use the Master theorem to provide an instant asymptotic solution

The Master Theorem: – Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n)$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = O(n^{\log_b a - \epsilon})$.

2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.

3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$, and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Examples of the Master Theorem

Which case of the Master Theorem applies?

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

  Reading from the equation, $a = 4$, $b = 2$, and $f(n) = n$.

  Is $n = O(n^{\log_2 4 - \epsilon}) = O(n^{2-\epsilon})$?

  Yes, so case 1 applies and $T(n) = O(n^2)$.

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

  Reading from the equation, $a = 4$, $b = 2$, and $f(n) = n^2$.

  Is $n^2 = O(n^{\log_2 4 - \epsilon}) = O(n^{2-\epsilon})$?

  No, if $\epsilon > 0$, but it is true if $\epsilon = 0$, so case 2 applies and $T(n) = \Theta(n^2 \log n)$.

- $T(n) = 4T(n/2) + n^3$

  Reading from the equation, $a = 4$, $b = 2$, and $f(n) = n^3$.

  Is $n^3 = \Omega(n^{\log_2 4 + \epsilon}) = \Omega(n^{2+\epsilon})$?

  Yes, for $0 < \epsilon 1$, so case 3 *might* apply.

  Is $4(n/2)^3 \le c \cdot n^3$?

  Yes, for $c \ge 1/2$, so there exists a $c < 1$ to satisfy the regularity condition, so case 3 applies and $T(n) = \Theta(n^3)$.