

This and other notes are available from <http://www.cs.berkeley.edu/~yozo/cs170.fa05/Master's Theorem>

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

We compare $f(n)$ to n^d where Let $d = \log_b a$.

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } f(n) = O(n^{d-\epsilon}) \text{ for some } \epsilon > 0 \\ \Theta(n^d \log n) & \text{if } f(n) = \Theta(n^d) \\ \Theta(f(n)) & \text{if } f(n) = \Omega(n^{d+\epsilon}) \text{ for some } \epsilon > 0 \text{ and} \\ & af\left(\frac{n}{b}\right) \leq cf(n) \text{ for some constant } c. \end{cases}$$

In particular, if $f(n) = n^k$:

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

Then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } k > \log_b a \\ \Theta(n^k \log n) & \text{if } k = \log_b a \\ \Theta(n^{\log_b a}) & \text{if } k < \log_b a \end{cases} .$$

Example: $T(n) = 4T(n/4) + n \log n$.

Here $d = \log_b a = 1$, but $n \log n \neq \Omega(n^{1+\epsilon})$ for any $\epsilon > 0$. Thus the master theorem does not apply. We try expanding $T(n)$:

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{4}\right) + n \log n \\ &= 4 \left[T\left(\frac{n}{4^2}\right) + \frac{n}{4} \log \frac{n}{4} \right] + n \log n \\ &= 4^2 \left[T\left(\frac{n}{4^3}\right) + \frac{n}{4^2} \log \frac{n}{4^2} \right] + n \log \frac{n}{4} + n \log n \\ &\quad \vdots \\ &= 4^k T\left(\frac{n}{4^k}\right) + n \sum_{i=0}^{k-1} \log \frac{n}{4^i} \end{aligned}$$

The above can be shown more rigorously by induction. Now let $k = \lfloor \log_4 n \rfloor$ so that $\frac{n}{4^k} \leq 4$. Hence $4^k = \Theta(n)$ and $T\left(\frac{n}{4^k}\right) = \Theta(1)$ (why?). Thus we have

$$T(n) = \Theta(n) + n \sum_{i=0}^{k-1} \log \frac{n}{4^i} = \Theta(n) + nS(n)$$

where

$$S(n) = \sum_{i=0}^{k-1} \log \frac{n}{4^i} = \sum_{i=0}^{k-1} (\log n - i) = k \log n - \frac{k(k-1)}{2} = \Theta(\log^2 n)$$

(Why does the above relation hold?) Thus we have $T(n) = \Theta(n \log^2 n)$.

Questions

1. Does the Master's theorem cover all cases? If not, give examples.

2. Give a tight bound for the recurrence

$$T(n) = 2T\left(\frac{n}{2}\right) + n \log^k n$$

for $k = \dots, -2, -1, 0, 1, 2, \dots$

3. Give a divide-and-conquer algorithm for summing an array of n numbers. What is the running time?
4. What is the running time of merge-sort if we divide the array into k pieces at each recursion step? Which k gives the fastest asymptotic running time?
5. Closest pair. Suppose you are given a set of n points p_1, p_2, \dots, p_n in the plane with coordinates $p_i = (x_i, y_i)$. Give a divide-and-conquer algorithm for finding the closest pair of points. What is the running time?