The Master Method

May 21, 2019

The "Master Theorem" provides a formula for the solution for many recurrence relations. Suppose that $a \geq 1$ and $b > 1$. Consider the recurrence

$$T(n) = aT(\frac{n}{b}) + \Theta(n^y)$$

in sloppy or exact form. Denote $x = log_b^a$. Then

$$T(n) \in \begin{cases} \Theta(n^x) & \text{if } y < x \\ \Theta(n^x logn) & \text{if } y = x \\ \Theta(n^y) & \text{if } y > x \end{cases}$$

The Master Theorem is really similar to Merge sort ($\Theta(nlogn)$).
Suppose that $a \geq 1$ and $b \geq 2$ are integers and

$$T(n) = aT(\frac{n}{b}) + cn^y, \quad T(1) = d$$

Table 1: Solution table:

Size of subproblem	**# nodes**	**cost/node**	**total cost**
$n = b^j$	1	cn^y	cn^y
$\frac{n}{b} = b^{j-1}$	a	$c(\frac{n}{b})^y$	$ca(\frac{n}{b})^y$
$\frac{n}{b^2} = b^{j-2}$	a^2	$c(\frac{n}{b^2})^y$	$ca^2(\frac{n}{b^2})^y$
$\cdots$	$\cdots$	$\cdots$	$cdots$
$\frac{n}{b^{j-1}} = b$	a^{j-1}	$c(\frac{n}{b^{j-1}})^y$	$ca^{j-1}(\frac{n}{b^{j-1}})^y$
$\frac{n}{b^j} = 1$	a^j	d	da^j

Proof:
Consider $a^j = (b^x)^j = (b^j)^x = n^x$
Then we have

$$da^j + cn^y \sum_{i=0}^{j-1} (\frac{a}{b^y})^i$$
$$= dn^x + cn^y \sum_{i=0}^{j-1} r^i \text{ where } r = \frac{a}{b^y}$$

Consider $r = \frac{a}{b^y} = \frac{b^x}{b^y} = b^{x-y}$
Let

$$S = \sum_{i=0}^{j-1} r^i \in \Theta(r^j)$$

Since $1 + r + r^2 + \cdots + r^{j-1} = \frac{r^j}{j-1} \in \Theta(r^j)$
Thus $r^j = (b^{x-y})^j = (b^j)^{x-y} = n^{x-y}$
Therefore we have

$$T(n) = dn^x + cn^y n^{x-y}$$

$$\begin{cases} \text{If } r > 1 \ (i.e. x > y) & \rightarrow \Theta(dn^x + cn^y n^{x-y}) \in \Theta(n^x) \\ \text{If } r = 1 \ (i.e. x = y) & \rightarrow \Theta(dn^x + cn^y \sum_{i=0}^{j-1} r^i) \in \Theta(dn^x + cn^y j) \\ & \rightarrow \Theta(dcj(n^x)) \in \Theta(n^x(logn)) \\ \text{If } r < 1 \ (i.e. x < y) & \rightarrow \Theta(dn^x + cn^y) \in \Theta(n^y) \end{cases}$$

Master Method Question & Explanation:

1. $T(n) = T(n-1) + n$

Consider: $T(n) = T(n-1) + n$

$= T(n-2) + n-1+n$

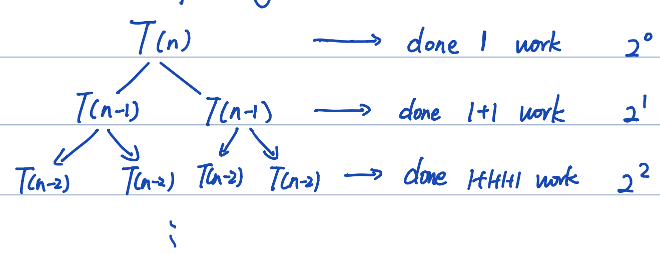
$= T(n-3) + n-2+n-1+n$

$\vdots$

$= T(1) + 2 + \cdots + n-1+n = \frac{(1+n)n}{2} \in \Theta(n^2)$

2. $T(n) = 2T(n-1) + 1$

Consider the following tree

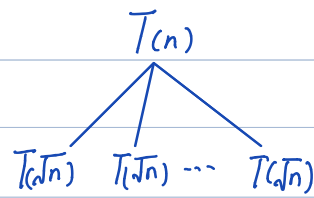


Thus we got $1+2+4+\cdots 2^{n-1} = \frac{1(2^n-1)}{2-1} = 2^n-1 \in \Theta(2^n)$

3. $T(n) = \sqrt{n}\, T(\sqrt{n}) + n \qquad (T(1) = 2)$

The question is hard, but we can apply the same method shown above.

Consider the following tree:



done n work

there are $\sqrt{n}$ nodes, and each nodes done $\sqrt{n}$ works.

total we have $\sqrt{n} \cdot \sqrt{n} = n$ work

for i^{th} level we have $n^{(\frac{1}{2})^i}$ nodes and each nodes done $\prod_{k=0}^{i} n^{(\frac{1}{2})^k} \cdot n^{(\frac{1}{2})^i}$

Therefore each level we do $n^{(\frac{1}{2})^i} \cdot n^{(\frac{1}{2})^i} \cdot \prod_{k=0}^{i} n^{(\frac{1}{2})^k} = n$ work

Consider the height of tree

$n^{(\frac{1}{2})^k} = 2$

$(\frac{1}{2})^k \log n = 1$

$\log n = 2^k$

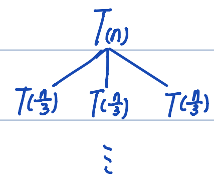
$\log\log n = k$

$\therefore$ we have $\log(\log n)$ levels and each level we do n works.

To sum up, we have $O(n \log\log n)$ complexity.

4. $T(n) = 3T(\frac{n}{3}) + \frac{n}{\log n}$

we also construct a tree to solve this type of problem



$\rightarrow$ done $\frac{n}{\log n}$ work

$\rightarrow$ done $3 \cdot \frac{n/3}{\log \frac{n}{3}} = \frac{n}{\log \frac{n}{3}} = \frac{n \log 3}{\log n}$ work

for i^{th} level we have 3^i nodes, and each nodes done $\frac{n/3^i}{\log n - \log 3^i}$ work

Therefore for i^{th} level. we have $3^i \cdot \frac{\frac{n}{3^i}}{\log n - \log 3^i} = \frac{n}{\log n - \log 3^i}$ work

Total we have $\log_3 n$ levels.

Then we have

$$T(n) = \frac{n}{\log n} + \frac{n}{\log n - \log 3} + \frac{n}{\log n - 2\log 3} + \cdots + \frac{n}{\log n - \log_3 n \cdot \log 3}$$

$$= \sum_{i=0}^{\log_3 n} \frac{n}{\log n - i\log 3} = \frac{n}{\log 3} \sum_{i=0}^{\log_3 n} \frac{1}{\log_3 n - i}$$

$$= \frac{n}{\log 3} \sum_{i=0}^{\log_3 n} \frac{1}{i}$$

$$= \frac{n}{\log 3} \cdot \log(\log_3 n) \in \Theta(n \log(\log_3 n))$$