Design and Analysis of Algorithms
Recurrence

Si Wu

School of CSE, SCUT

cswusi@scut.edu.cn

TA: Wenhao Wu (1565865638@qq.com)
Yi Liu (1337545838@qq.com)

Topics

- **Master Method**

Master Method

Goal. Recipe for solving common divide-and-conquer recurrences:

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

With $T(0) = 0 \; and \; T(1) = \Theta(1)$.

Terms.

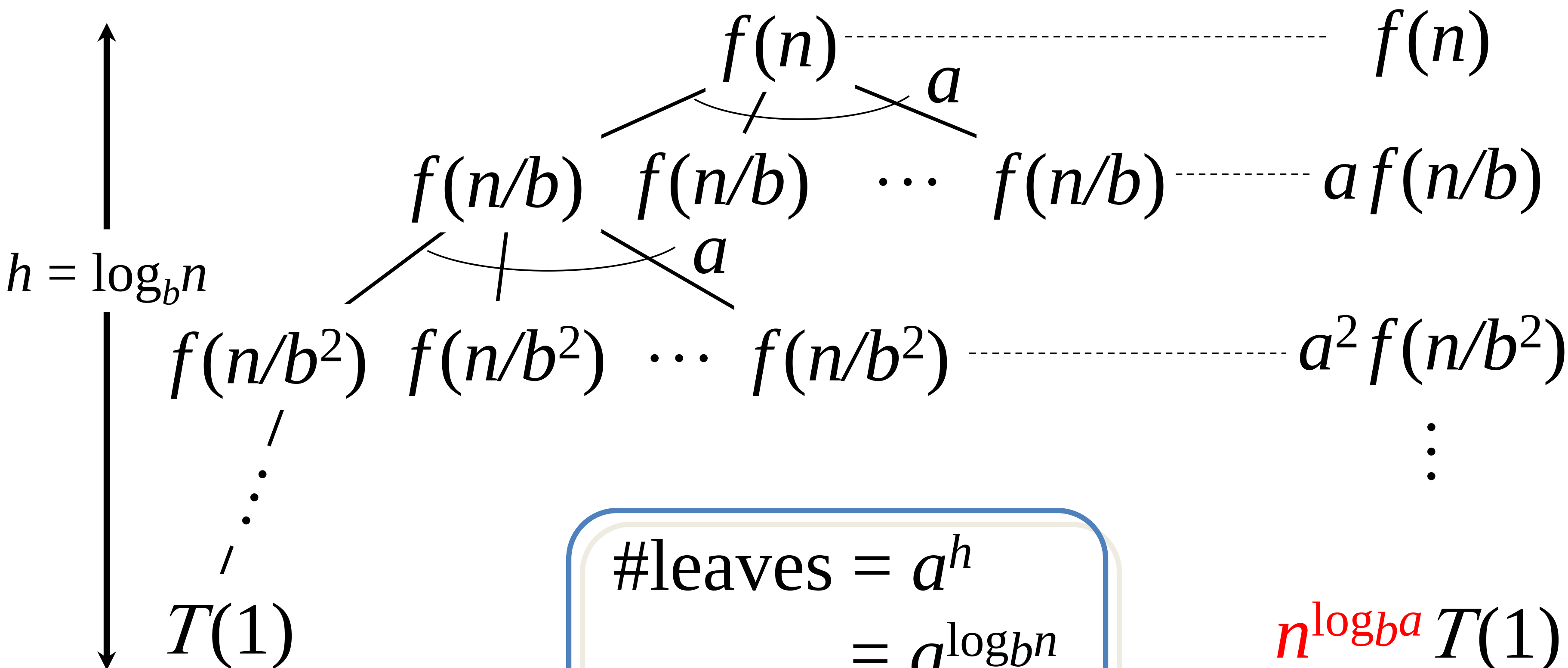
- $a \geq 1$ is the (integer) number of subproblems.
- $b > 1$ is the (integer) factor by which the subproblem size decreases.
- $f(n) =$ work to divide and combine subproblems.

Recursion tree.

- Number of levels:
- Number of subproblems at level i:
- Size of subproblem at level i:
- Number of leaves:

Idea of Master Theorem

Build a recursion tree:



$$\begin{aligned}\#\text{leaves} &= a^h \\ &= a^{\log_b n} \\ &= n^{\log_b a}\end{aligned}$$

Master Method

Goal. Recipe for solving common divide-and-conquer recurrences:

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

With $T(0) = 0$ and $T(1) = \Theta(1)$.

Terms.

- $a \geq 1$ is the (integer) number of subproblems.
- $b > 1$ is the (integer) factor by which the subproblem size decreases.
- $f(n) =$ work to divide and combine subproblems.

Recursion tree.

- Number of levels: $k = \log_b n$.
- Number of subproblems at level i: a^i.
- Size of subproblem at level i: n/b^i.
- Number of leaves: $n^{\log_b a}$.

Master Theorem

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

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

with $T(0) = 0 \ and \ T(1) = \Theta(1)$, where n/b means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then,

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right).$

Ex. $T(n) = 3T(n/2) + 5n$
$a = 3, b = 2, f(n) = 5n, k = 1, \log_b a = 1.58$
$T(n) = \Theta\left(n^{\log_2 3}\right)$

Master Theorem

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

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

with $T(0) = 0 \; and \; T(1) = \Theta(1)$, where n/b means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then,

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Ex. $T(n) = 2T(n/2) + 17n \log n$

$a = 2, b = 2, f(n) = 17n \log n \,, k = 1, p = 1, \log_b a = 1$

$T(n) = \Theta\left(n \, log^2 \, n\right)$

Master Theorem

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

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

with $T(0) = 0\ and\ T(1) = \Theta(1)$, where n/b means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then,

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

Ex. $T(n) = 3T(n/2) + n^2$
$a = 3, b = 2, f(n) = n^2, k = 2, \log_b a = 1.58$
Regularity condition: $3(n/2)^2 \leq cn^2$ for $c = 3/4$
$T(n) = \Theta(n^2)$

Master Theorem

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

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

with $T(0) = 0 \; and \; T(1) = \Theta(1)$, where n/b means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$.

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.
Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.
Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta\left(f(n)\right)$.

Proof Sketch.

- Use recursion tree to sum up terms (assuming n is an exact power of b)
- Three cases for geometric series.

Master Theorem Need Not Apply

Gaps in master theorem

- Number of subproblems must be a constant.

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

- Number of subproblems must be ≥ 1.

$$T(n) = \frac{1}{2}T(n/2) + n^2$$

- Non-polynomial separation between $f(n)$ and $\log n$.

$$T(n) = 2T(n/2) + \frac{n}{logn}$$

- $f(n)$ is not positive.

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

- Regularity condition does not hold.

$$T(n) = T(n/2) + \mathrm{n}(2 - \cos n)$$