Lecture 4b: The Master Theorem

A General Technique for solving
Divide-and-Conquer Recurrences

Version of February 18, 2019

Outline

- Introduction to Divide-and-Conquer Recurrences
- The Master Theorem Statement
- Derivation of Master Theorem when $f(n) = \theta(n)$

Divide-and-Conquer

Already saw a few divide-and-conquer algorithms

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

Divide

- Divide a given problem into a or more subproblems (ideally of approximately equal size n/b)

Conquer $a \cdot T(n/b)$

- Solve each subproblem (directly if small enough or recursively)

Combine $f(n)$

- Combine the solutions of the subproblems into a global solution

Divide-and-Conquer Examples

Four major examples so far

- Maximum Contiguous Subarray & Mergesort
  - Both satisfied $T(n) = 2T(n/2) + O(n)$
  - $T(n) = O(n \log n)$

- First version of Polynomial Multiplication
  - $T(n) = 4T(n/2) + O(n)$
  - $T(n) = O(n^2)$

- Karatsuba Multiplication
  - $T(n) = 3T(n/2) + O(n)$
  - $T(n) = O\left(n^{\log_2 3}\right) = O(n^{1.58\ldots})$

Outline

- Introduction to Divide-and-Conquer Recurrences
- The Master Theorem Statement
- Derivation of Master Theorem when $f(n) = \theta(n)$

The Master Theorem

Main tool is the Master Theorem for solving recurrences of form

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

where

- $a \geq 1$ and $b > 1$ are constants and
- $f(n)$ is a (asymptotically) positive function.
- Note: Initial conditions are $T(1), T(2), \ldots, T(k)$ for some k. They don't contribute to asymptotic growth
- n/b could be either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$

The Master Theorem (for equalities)

$T(n) = aT(n/b) + f(n), \qquad c = \log_b a$

1. If $f(n) = \theta(n^{c-\epsilon})$ for some $\epsilon > 0$ => $\mathrm{T}(n) = \theta(n^c)$

2. If $f(n) = \theta(n^c)$ => $\mathrm{T}(n) = \theta(n^c \log n)$

3. If $f(n) = \theta(n^{c+\epsilon})$ for some $\epsilon > 0$
   and if $af(n/b) \leq df(n)$
   for some $d < 1$ and large enough n
   => $\mathrm{T}(n) = \theta(f(n))$

The Master Theorem (for inequalities)

$T(n) \leq aT(n/b) + f(n), \qquad c = \log_b a$

1. If $f(n) = O(n^{c-\epsilon})$ for some $\epsilon > 0$ => $\mathrm{T}(n) = O(n^c)$

2. If $f(n) = O(n^c)$ => $\mathrm{T}(n) = O(n^c \log n)$

3. If $f(n) = O(n^{c+\epsilon})$ for some $\epsilon > 0$
   and if $af(n/b) \leq df(n)$
   for some $d < 1$ and large enough n

   => $\mathrm{T}(n) = O(f(n))$

The Master Theorem when $f(n) = \theta(n)$

$T(n) = aT(n/b) + f(n), \qquad c = \log_b a$

Note: Inequality version of theorem also holds

1. If $c > 1$, then $T(n) = \theta(n^c)$
   - If $T(n) = 4T(n/2) + \theta(n)$ then $T(n) = \theta(n^2)$
   - If $T(n) = 3T(n/2) + \theta(n)$ then $T(n) = \theta\left(n^{\log_2 3}\right) = \theta(n^{1.58\ldots})$

2. If $c = 1$, then $T(n) = \theta(n \log n)$
   - If $T(n) = 2T(n/2) + \theta(n)$ then $T(n) = \theta(n \log n)$

3. If $c < 1$, then $T(n) = \theta(n)$
   - If $T(n) = T(n/2) + \theta(n)$. then $T(n) = \theta(n)$

More Master Theorem(s)

There are many variations of the Master Theorem.

Here's another...

- If $T(n) = T(3n/4) + T(n/5) + n$ then

$$T(n) = \theta(n)$$

- More generally, given constants $\alpha_i > 0$ with $\sum_i \alpha_i < 1$

  If $T(n) = n + \sum_i T(\alpha_i n)$ then

$$T(n) = \theta(n)$$

Outline

- Introduction to Divide-and-Conquer Recurrences
- The Master Theorem Statement
- Derivation of Master Theorem when $f(n) = \theta(n)$

The Master Theorem when $f(n) = O(n)$

$T(n) \le aT(n/b) + kn,$ $a \ge 1$ and b > 1 are constants, $c = \log_b a$

1. If $\mathrm{c} > 1,$ then $T(n) = O(n^c)$

2. If $\mathrm{c} = 1,$ then $T(n) = O(n \log n)$

3. If $\mathrm{c} < 1,$ then $T(n) = O(n)$

Have already worked through two examples of case 1 and one example of case 2.
Will now see general proof.

Proof of Inequality Master Theorem when $f(n) = O(n)$

$T(n) \leq aT(n/b) + kn,$ $a \geq 1$ and b > 1 are constants, $c = \log_b a$

Assume $n = b^h$

$$T(n) \leq a\,T\left(\frac{n}{b}\right) + kn$$

$$\leq a\left[a\,T\left(\frac{n}{b^2}\right) + k\frac{n}{b}\right] + kn$$

$$= a^2 T\left(\frac{n}{b^2}\right) + \left(1 + \frac{a}{b}\right) kn$$

$$\leq a^2\left[a\,T\left(\frac{n}{b^3}\right) + k\frac{n}{b^2}\right] + \left(1 + \frac{a}{b}\right) kn$$

$$= a^3 T\left(\frac{n}{b^3}\right) + \left(1 + \frac{a}{b} + \left(\frac{a}{b}\right)^2\right) kn$$

.....

$$\leq a^h T\left(\frac{n}{b^h}\right) + \sum_{j=0}^{h-1}\left(\frac{a}{b}\right)^j kn$$

We now examine each of the three cases of c separately

$\boxed{c = \log_b a}$

Case 1: $a > b \quad (c > 1)$

Assume $n = b^h$. Then

$$a^h = \left(b^{\log_b a}\right)^h = b^{h \log_b a} = \left(b^h\right)^{\log_b a} = n^{\log_b a} = n^c$$

If $a > b$

$$\sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j = \frac{\left((a/b)^h - 1\right)}{a/b \ - 1} \le \frac{(a/b)^h}{a/b - 1} = \frac{n^{\log_b a}/n}{a/b - 1} = \frac{n^c/n}{a/b - 1} = \frac{n^{c-1}}{a/b - 1}$$

Recall that $T(n) \le a^h T\left(\frac{n}{b^h}\right) + \sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j \ kn.$

Hence

$$\boxed{T(n) = O\left(\underbrace{n^c}_{a^h} \ \underbrace{T(1)}_{T\left(\frac{n}{b^h}\right)} + \underbrace{\frac{n^{c-1}}{a/b - 1}}_{\sum_{j=0}^{h-1}\left(\frac{a}{b}\right)^j} kn\right) = O(n^c)}$$

$\boxed{c = \log_b a}$

Case 1: $a > b \quad (c > 1)$

Assume $n = b^h$. Then

$$a^h = \left(b^{\log_b a}\right)^h = b^{h \log_b a} = \left(b^h\right)^{\log_b a} = n^{\log_b a} = n^c$$

If $a > b$

$$\sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j = \frac{\left((a/b)^h - 1\right)}{a/b \ - 1} \le \frac{(a/b)^h}{a/b - 1} = \frac{n^{\log_b a}/n}{a/b - 1} = \frac{n^c/n}{a/b - 1} = \frac{n^{c-1}}{a/b - 1}$$

Recall that $T(n) \le a^h T\left(\frac{n}{b^h}\right) + \sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j \ kn.$

Hence

$$\boxed{T(n) = O\left(n^c \, T(1) + \frac{n^{c-1}}{a/b - 1} kn\right) = O(n^c)}$$

Example: If $T(n) \le 3T\left(\frac{n}{2}\right) + n$ then $a = 3,\ \mathrm{b} = 2$

$$=> T(n) = O\left(n^{\log_2 3}\right) = O(n^{1.58\ldots})$$

$c = \log_b a$

Case 2: $a = b \quad (c = 1)$

Assume $n = b^h$. Then

$$a^h = \left(b^{\log_b a}\right)^h = b^{h \log_b a} = \left(b^h\right)^{\log_b a} = n^{\log_b a} = n^c = n$$

If $a = b$

$$\sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j = \sum_{j=0}^{h-1} 1^j = h$$

Hence

$$T(n) \leq a^h T\left(\frac{n}{b^h}\right) + \sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j \; kn = O(n + hn) = O(n \log n)$$

Example: If $T(n) \leq 2T\left(\frac{n}{2}\right) + n$ then $a = 2$, $\mathrm{b} = 2$

$\Rightarrow T(n) = O(n \log n)$

$c = \log_b a$

Case 3: $a < b \quad (c < 1)$

Assume $n = b^h$. Then

$$a^h = \left(b^{\log_b a}\right)^h = b^{h\log_b a} = \left(b^h\right)^{\log_b a} = n^{\log_b a} = n^c = O(n)$$

If $a < b$

$$\sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j = \frac{\left((a/b)^h - 1\right)}{a/b \; - 1} = \frac{\left(1 - (a/b)^h\right)}{1 - a/b} = O(1)$$

Hence

$$T(n) \le a^h T\left(\frac{n}{b^h}\right) + \sum_{j=0}^{h-1} \left(\frac{a}{b}\right)^j kn = O\left(n^{\log_b a} + n\right) = O(n)$$

Example: If $T(n) \le 2T\left(\frac{n}{3}\right) + n$ then $a = 2,\ \mathrm{b} = 3$

$=> T(n) = O(n)$

Sometimes known as *Decimation*

Reprise: Most Useful Master Theorem (for inequalities)

$T(n) \leq aT(n/b) + f(n), \qquad c = \log_b a$

1. If $f(n) = O(n^{c-\epsilon})$ for some $\epsilon > 0$ then $\mathrm{T}(n) = O(n^c)$

2. If $f(n) = O(n^c)$ then $\mathrm{T}(n) = O(n^c \log n)$

3. If $f(n) = O(n^{c+\epsilon})$ for some $\epsilon > 0$
   and if $af(n/b) \leq df(n)$
   for some $d < 1$ and large enough n
   then $\mathrm{T}(n) = O(f(n))$