4.3 - The master method

- The master method provides a cookbook for solving recurrences of the form

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n) \quad \text{where } \underline{a \geq 1} \text{ and } \underline{b > 1}$$

- Requires memorization of three cases

- Recurrence describes the running time of an algorithm that:
  - divides a problem of size $\underline{n}$ into $\underline{a}$ subproblems;
  - each subproblem has input of size n/b
  - ($\underline{a}$ and $\underline{b}$ are positive constants;)
  - The $\underline{a}$ subproblems are solved recursively. Each one takes time $T(n/b)$
  - The cost of dividing the problem and combining the results is described by function $f(n)$
  - MergeSort: $T(n) = \underbrace{2}_{a} T(n/\underbrace{2}_{b}) + \underbrace{\Theta(n)}_{f(n)}$
  - Again: We omit floor and ceiling functions when writing recurrences of this form.

4.3.1 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:

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

Then $T(n)$ can be bounded asymptotically as follows:

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

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

Case 3. If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$ and if $a f\left(\frac{n}{b}\right) \leq c f(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$ ∎

Question:
What does the theorem say?

- We are comparing $f(n)$ with $n^{\log_b a}$
- Intuitively: the solution to the recurrence is determined by the larger of the two functions
  - If function $n^{\log_b a}$ is larger (case 1) then $T(n) = \Theta(n^{\log_b a})$
  - If function $f(n)$ is larger (case 3) then $T(n) = \Theta(f(n))$
  - If both functions are the same size (case 2) then $T(n) = \Theta(n^{\log_b a} \times \log n)$

→

Some important technicalities:

Case 1: Not only must $f(n)$ be smaller than $n^{\log_b a}$ it must be polynomially smaller (i.e. a factor of n^{ε}) $\varepsilon > 0$

Case 3: Not only must $f(n)$ be larger $n^{\log_b a}$ it must be polynomially larger (i.e. a factor of n^{ε}, $\varepsilon > 0$)

and satisfy $a\, f\left(\frac{n}{b}\right) \le c\, f(n)$

Example:

- $T(n) = 9\,T\left(\frac{n}{3}\right) + n$

$$\left.\begin{array}{l} a = 9 \\ b = 3 \\ f(n) = n \end{array}\right\} \underbrace{\log_b a = \log_3 9 = 2}$$

$\Downarrow$ Case 1

$$f(n) = O(n^{\log_b a - \varepsilon}) = O(n^{2-\varepsilon}) \quad \varepsilon = 1 \Rightarrow T(n) = \Theta(n^2)$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Example:

- $T(n) = T\left(\frac{2n}{3}\right) + 1$

$$\left.\begin{array}{l} a = 1 \\ b = 3/2 \\ f(n) = 1 \end{array}\right\} \underbrace{\log_b a = \log_{3/2} 1 = 0}$$

$\Downarrow$ Case 2

$$f(n) = \Theta(n^{\log_b a}) = \Theta(n^0) = \Theta(1) \Rightarrow T(n) = \Theta(n^0 \log n) = \Theta(\log n)$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = f(n)$

Example:

- $T(n) = 3\,T\left(\frac{n}{4}\right) + n \log n$

$$\left.\begin{array}{l} a = 3 \\ b = 4 \\ f(n) = n \log n \end{array}\right\} \underbrace{n^{\log_b a} = n^{\log_4 3} = n^{0,792\ldots}}$$

$\Downarrow$ Case 3

$$\underbrace{f(n) = \Omega(n^{\log_b a + \varepsilon}) = \Omega(n^{0,792\ldots + \varepsilon}), \varepsilon \simeq 0,21\ldots}$$

$\Downarrow$

$$T(n) = f(n) = n \log n$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Example:

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

$$\left.\begin{array}{l} a = 2 \\ b = 2 \\ f(n) = n \log n \end{array}\right\} \underbrace{\log_b a = \log_2 2 = 1}$$

Seems like case 3, right?

Wrong!!!

Ratio: $\frac{f(n)}{n^{\log_b a}} = \frac{n \log n}{n} = \log n$ Not polynomially larger!!!

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Example:

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

$a = 4$, $b = 2$, $f(n) = n$

$$\underbrace{\log_b a = \log_2 4 = 2}$$

case 1:

$$\underbrace{f(n) = O(n^{\log_b a - \varepsilon}) = O(n^{2-\varepsilon}),\ \varepsilon = 1}$$

$$\Downarrow$$

$$T(n) = \Theta(n^2)$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Example:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$a = 4$, $b = 2$, $f(n) = n^2$

$$\underbrace{\log_b a = \log_2 4 = 2 \Rightarrow n^{\log_b a} = n^2}_{\text{Case 2}}$$

$$f(n) = \Theta(n^{\log_b a}) = \Theta(n^2) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n^2 \log n)$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Example:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

$a = 4$, $b = 2$, $f(n) = n^3$

$$\underbrace{\log_b a = \log_2 4 = 2 \Rightarrow n^{\log_b a} = n^2}_{\text{Case 3}}$$

$$\underbrace{f(n) = \Omega(n^{\log_b a + \varepsilon}) = \Omega(n^{2+\varepsilon}),\ \varepsilon = 1}$$

$$T(n) = \Theta(n^3)$$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

Exercises:

- $T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$ Answer: $T(n) = \Theta(\log n)$
- $T(n) = 4T\left(\frac{n}{2}\right) + n^2 \log n$ Question: polynomically larger/smaller?
- $T(n) = 2T\left(\frac{n}{2}\right) + n^3$
- $T(n) = T\left(\frac{9n}{10}\right) + n$
- $T(n) = 16T\left(\frac{n}{4}\right) + n^2$
- $T(n) = 7T\left(\frac{n}{3}\right) + n^2$
- $T(n) = 7T\left(\frac{n}{2}\right) + n^2$
- $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$
- $T(n) = T(n-1) + n$
- $T(n) = T(\sqrt{n}) + 1$

Question: Is in the form: $T(n) = aT\left(\frac{n}{b}\right) + f(n)$?