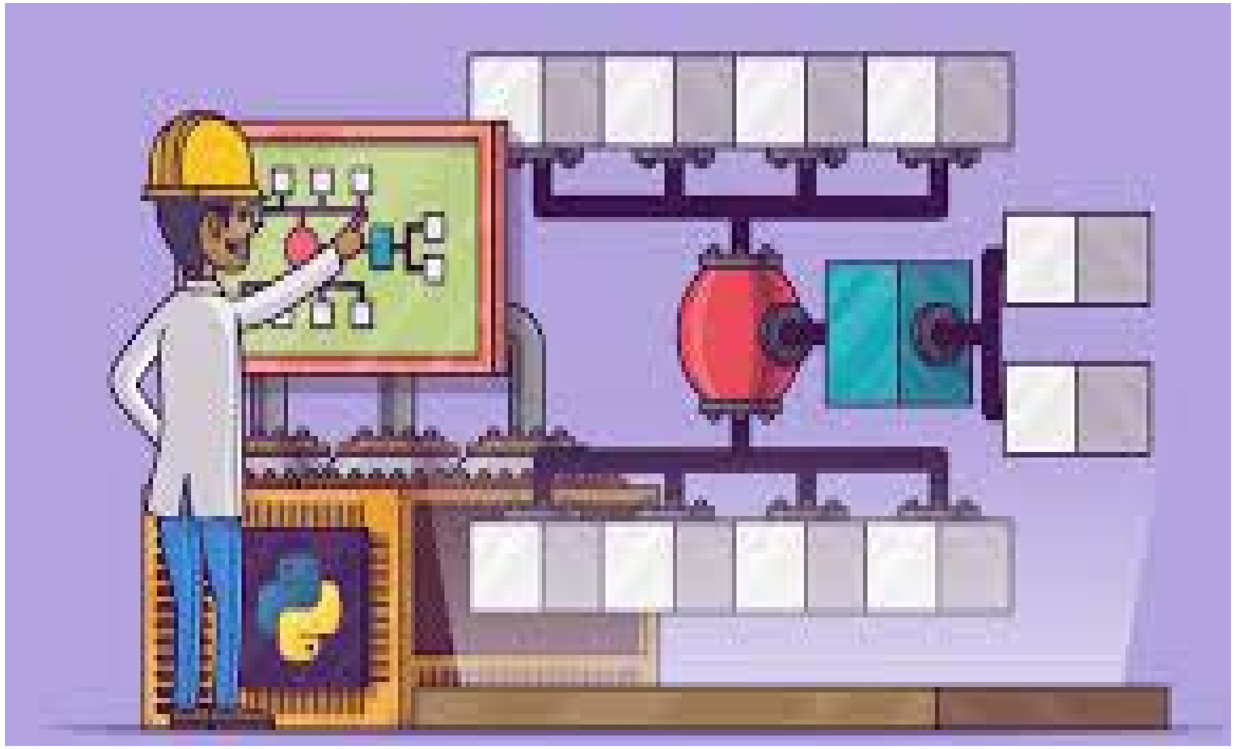
ساختمان داده ها



مدرس:
سمانه حسینی سمنانی

دانشگاه صنعتی اصفهان- دانشکده برق و کامپیوتر

The master method

$$T(n) = a\,T(n/b) + f(n)$$

$a \geq 1$ and $b > 1$

- $b < 1$
- $b = 1$

The master method

$$T(n) = a\,T(n/b) + f(n)$$

$$a \geq 1 \text{ and } b > 1$$

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

The master method

$$T(n) = a\,T(n/b) + f(n)$$

$a \geq 1$ and $b > 1$

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

The master method

$$T(n) = a\,T(n/b) + f(n)$$

$$a \geq 1 \text{ and } b > 1$$

3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■

The master method

$$T(n) = a\,T(n\,/\,b) + f(n)$$

$a \geq 1$ and $b > 1$

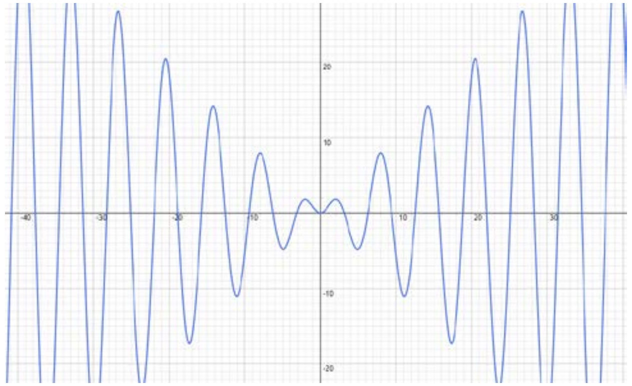
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $a f(n/b) \leq c f(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■

The master theorem

- In each of the three cases, we compare the function $f\ (n)$ with the function $n^{log_b^a}$
- the larger of the two functions determines the solution to the recurrence.
- If, as in case 1, the function $n^{log_b^a}$ is the larger, then the solution is $T(n) = \Theta(n^{\log_b a})$
- If, as in case 3, the function $f\ (n)$ is the larger, then the solution is $T(n) = \Theta(f(n))$.
- If, as in case 2, the two functions are the same size, we multiply by a logarithmic factor, and the solution is $T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(f(n) \lg n)$.

Can we use master theorem for any $f(n)$?

- $f(n) = o(n^{log_b^a})$ or
- $f(n) = \theta(n^{log_b^a})$ or
- $f(n) = \omega(n^{log_b^a})$
- Is this true for any $f(n)$? e.g. $f(n) = n.sin(n)$



- for ordinary cases that is True. But the problem is that master theorem does not use above equations.
- In master theorem the difference between $n^{log_b^a}$ and $f(n)$ should be polynomial
- it does not work for other functions e.g. $f(n) = loglogn$

The master theorem: Example 1

$$T(n) = 9T(n/3) + n$$

- $a = 9,$
- $b = 3,$
- $f(n) = n$
- $f(n) = O(n^{log_3^9 - \epsilon})$
- $T(n) = \theta(n^2)$ Case 1

The master theorem: Example 2

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

- $a = 1,$
- $b = 3/2,$
- $f(n) = 1$
- $f(n) = \theta(n^{log_{3/2}^{1}})$
- $T(n) = \theta(logn)$ Case 2

The master theorem: Example 3

$$T(n) = 3T(n/4) + n \lg n$$

- $a = 3,$
- $b = 4,$
- $f(n) = nlogn$
- $f(n) = \Omega\left(n^{log_4^3 + \epsilon}\right)$, $af(n/b) = 3(n/4)\lg(n/4) \le (3/4)n \lg n = cf(n)$
- $T(n) = \theta(nlogn)$ Case 3

The master theorem: Example 4

$$T(n) = 2T(n/2) + n \lg n$$

- $a = 2,$
- $b = 2,$
- $f\ (n) = nlogn$
- $f(n) = ?\left(n^{log_2^2+\epsilon}\right),$
- You might mistakenly think that case 3 should apply, since $f(n) = n \lg n$ is asymptotically larger than n.
- The problem is that it is not polynomially larger. $f(n)/n^{\log_b a} = (n \lg n)/n = \lg n$

The master theorem: Example 4

$$\text{if } f(n) = \Theta(n^{\log_b a} \lg^k n), \text{ where } k \geq 0,$$

then

$$T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$$

General form of Case 2

In case 2: k=0

The master theorem: Example 4

$$T(n) = 2T(n/2) + n \lg n$$

- $a = 2,$
- $b = 2,$
- $f(n) = nlogn$
- $f(n) = \theta\left(n^{log_2^2} log^k n\right), k = 1$
- $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n) = \theta\left(n^{log_2^2} log^2 n\right)$

Proof of Master theorem-*Lemma* 4.2

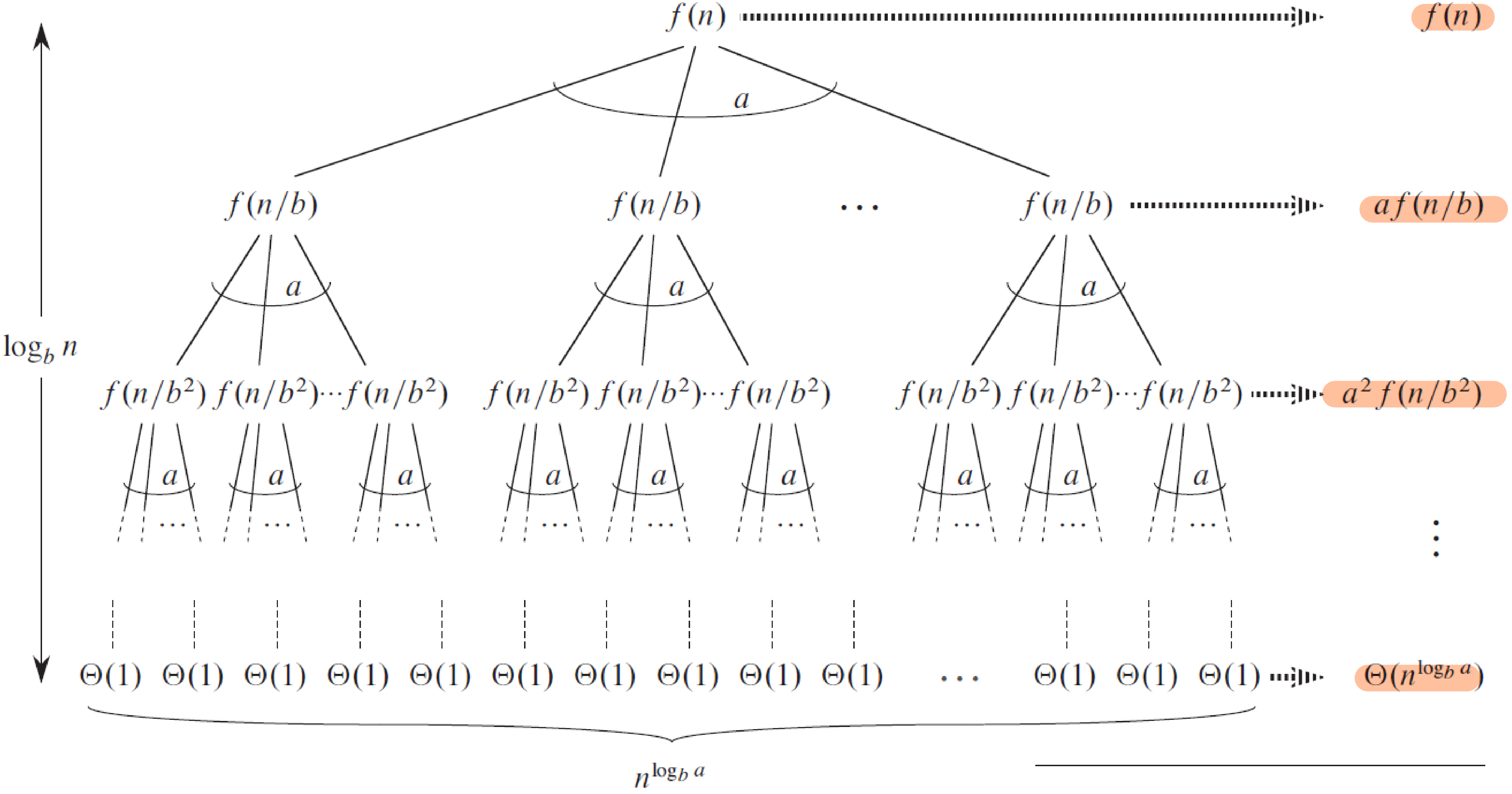
Lemma 4.2

$n = 1, b, b^2, \ldots.$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ aT(n/b) + f(n) & \text{if } n = b^i \end{cases}$$

then

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$



Total: $\Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$

Proof of Master theorem

- In terms of the recursion tree, the three cases of the master theorem correspond to cases in which the total cost of the tree is :

  (1) dominated by the costs in the leaves

  (2) evenly distributed among the levels of the tree

  (3) dominated by the cost of the root.

Proof of Master theorem-*Lemma* 4.3

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

has the following asymptotic bounds for exact powers of b:

1. If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant $\epsilon > 0$, then $g(n) = O(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $g(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $af(n/b) \le cf(n)$ for some constant $c < 1$ and for all sufficiently large n then $g(n) = \Theta(f(n))$.

Proof of Master theorem-*Lemma* 4.3

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

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

$$f(n) = O(n^{\log_b a-\epsilon}) \Rightarrow f(n/b^j) = O((n/b^j)^{\log_b a-\epsilon}) \Rightarrow$$

$$g(n) = O\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\epsilon}\right)$$

Proof of Master theorem-*Lemma* 4.3

$$\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\epsilon} = n^{\log_b a-\epsilon} \sum_{j=0}^{\log_b n-1} \left(\frac{ab^\epsilon}{b^{\log_b a}}\right)^j$$

$$= n^{\log_b a-\epsilon} \sum_{j=0}^{\log_b n-1} (b^\epsilon)^j$$

$$= n^{\log_b a-\epsilon} \left(\frac{b^{\epsilon \log_b n} - 1}{b^\epsilon - 1}\right)$$

$$= n^{\log_b a-\epsilon} \left(\frac{n^\epsilon - 1}{b^\epsilon - 1}\right)$$

Proof of Master theorem-*Lemma* 4.3

$$n^{\log_b a-\epsilon}\left(\frac{n^{\epsilon}-1}{b^{\epsilon}-1}\right) = n^{\log_b a-\epsilon}O(n^{\epsilon}) = O(n^{\log_b a}) \Longrightarrow$$

$$g(n) = O(n^{\log_b a})$$

Proof of Master theorem-*Lemma* 4.3

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

2. If $f(n) = \Theta(n^{\log_b a})$, then $g(n) = \Theta(n^{\log_b a} \lg n)$.

$$f(n) = \Theta(n^{\log_b a}) \Longrightarrow f(n/b^j) = \Theta((n/b^j)^{\log_b a}) \Longrightarrow$$

$$g(n) = \Theta\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

Proof of Master theorem-*Lemma* 4.3

$$
\begin{aligned}
\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a} &= n^{\log_b a} \sum_{j=0}^{\log_b n-1} \left(\frac{a}{b^{\log_b a}}\right)^j \\
&= n^{\log_b a} \sum_{j=0}^{\log_b n-1} 1 \\
&= n^{\log_b a} \log_b n \, .
\end{aligned}
$$

Proof of Master theorem-*Lemma* 4.3

$$\begin{aligned} g(n) &= \Theta(n^{\log_b a} \log_b n) \\ &= \Theta(n^{\log_b a} \lg n) \,, \end{aligned}$$

$b > 1$

Proof of Master theorem-*Lemma* 4.3

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

3. If $af(n/b) \le cf(n)$ for some constant $c < 1$ and for all sufficiently large then $g(n) = \Theta(f(n))$.

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j) \xrightarrow{a,b\ n,j \text{ are positive}} g(n) = \Omega(f(n))$$

$$af(n/b) \le cf(n) \Longrightarrow f(n/b) \le (c/a)f(n) \xrightarrow{?} f(n/b^j) \le (c/a)^j f(n)$$

Proof of Master theorem-*Lemma* 4.3

$$af(n/b) \le cf(n) \Longrightarrow f(n/b) \le (c/a)f(n) \overset{?}{\Longrightarrow} f(n/b^j) \le (c/a)^j f(n)$$

$$f(n/b) \le (c/a)f(n)$$

$$f(n/b^2) \le (c/a)f(n/b) \le (c/a)^2 f(n)$$

$$\vdots$$

$$f(n/b^j) \le (c/a)^j f(n) \Longrightarrow$$

$$a^j f(n/b^j) \le c^j f(n)$$

Proof of Master theorem-*Lemma* 4.3

$$
\begin{aligned}
g(n) &= \sum_{j=0}^{\log_b n-1} a^j f(n/b^j) \\
&\le \sum_{j=0}^{\log_b n-1} c^j f(n) + O(1) \\
&\le f(n) \sum_{j=0}^{\infty} c^j + O(1) \\
&= f(n)\left(\frac{1}{1-c}\right) + O(1) \\
&= O(f(n)) , \quad g(n) = \Omega(f(n)) \Longrightarrow g(n) = \Theta(f(n))
\end{aligned}
$$

Proof of Master theorem

$$T(n) = a\,T(n\,/\,b) + f(n)$$

$a \geq 1$ and $b > 1$

1. If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■

Proof of Master theorem

from lemma 4-2:
$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

from lemma 4-3:
$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

1. If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant $\epsilon > 0$, then $g(n) = O(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $g(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $af(n/b) \leq cf(n)$ for some constant $c < 1$ and for all sufficiently large n then $g(n) = \Theta(f(n))$.

Proof of Master theorem

- Case 1:

$$\begin{aligned} T(n) &= \Theta(n^{\log_b a}) + O(n^{\log_b a}) \\ &= \Theta(n^{\log_b a}) , \end{aligned}$$

- Case 2:

$$\begin{aligned} T(n) &= \Theta(n^{\log_b a}) + \Theta(n^{\log_b a} \lg n) \\ &= \Theta(n^{\log_b a} \lg n) . \end{aligned}$$

- Case 3:

$$\begin{aligned} T(n) &= \Theta(n^{\log_b a}) + \Theta(f(n)) \\ &= \Theta(f(n)) , \end{aligned}$$

because $f(n) = \Omega(n^{\log_b a+\epsilon})$.

Proof of Master theorem for exact n

Lemma 4.2

$n = 1, b, b^2, \ldots.$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ T(n) = aT(\lceil n/b \rceil) + f(n) & \end{cases}$$

then

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

Proof of Master theorem for exact n

- First approach:
- Guess the solution for $n = 1, b, b^2, \ldots.$
- Then use Substitution method for exact n