Master Theorem

Divide-and-Conquer Recurrence Examples

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 Integer Multiplication
  - $T(n) = 4T(n/2) + O(n)$
  - $T(n) = O(n^2)$
- Karatsuba Multiplication
  - $T(n) = 3T(n/2) + O(n)$
  - $T(n) = O\left(n^{\log_2 3}\right) = O(n^{1.58\ldots})$

The Master Theorem

Tool for directly (i.e., without expansion or recurrence tree) solving recurrences of form

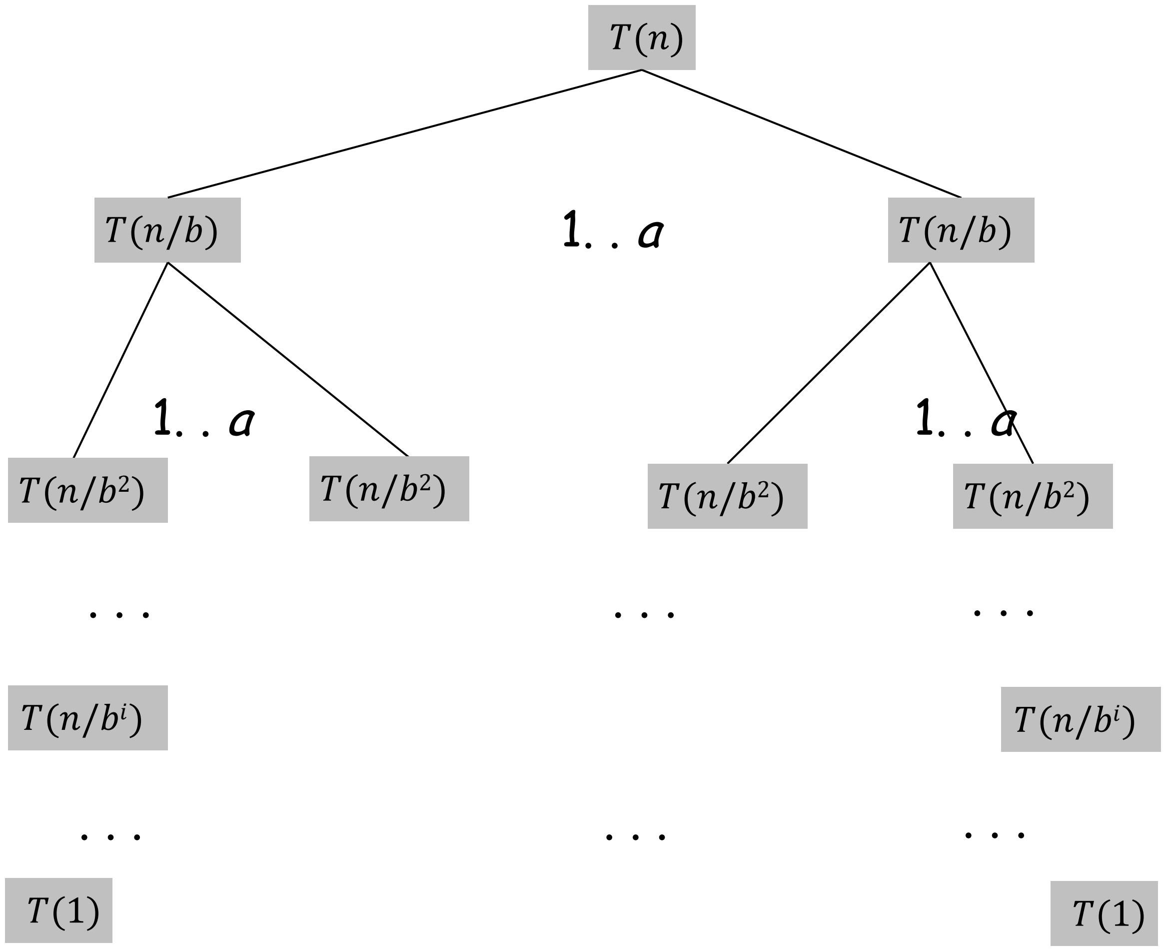
$$T(n) = aT(n/b) + f(n)$$

where

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

Visualization for Master Theorem

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$, $a \geq 1$ and $b > 1$ are constants, $c = \log_b a$



Lv	#pr	work/pr	work/lv
0	1	$f(n)$	$f(n)$
1	a	$f(n/b)$	$af(n/b)$
2	a^2	$f(n/b^2)$	$a^2 f(n/b^2)$
i	a^i	$f(n/b^i)$	$a^i f(n/b^i)$
$\log_b n$	$a^{\log_b n} = n^{\log_b a}$ $= n^c$	1	n^c

The Master Theorem (for equalities)

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

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

Intuition: the work increases as we go down the levels. Bottom level dominates the total cost.

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

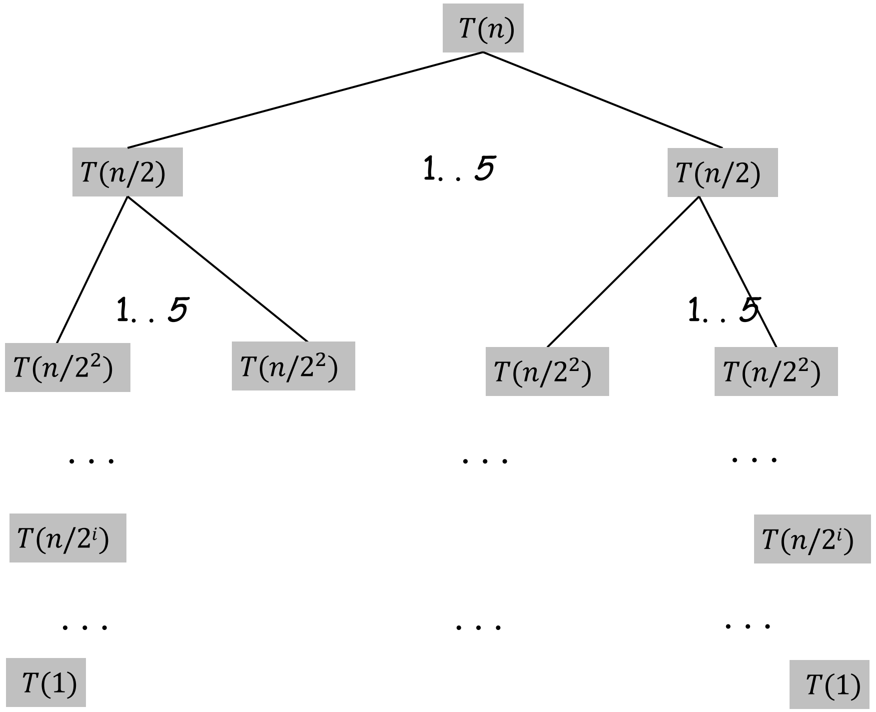
Intuition: the work remains the same as we go down the levels. All levels contribute equally to the total cost.

3. If $f(n) = \Omega(n^{c+\epsilon})$ for some $\epsilon > 0$ => $\mathrm{T}(n) = \Theta(f(n))$

(there is an additional condition that we will ignore in this class)

Intuition: the work decreases as we go down the levels. Top level dominates the total cost.

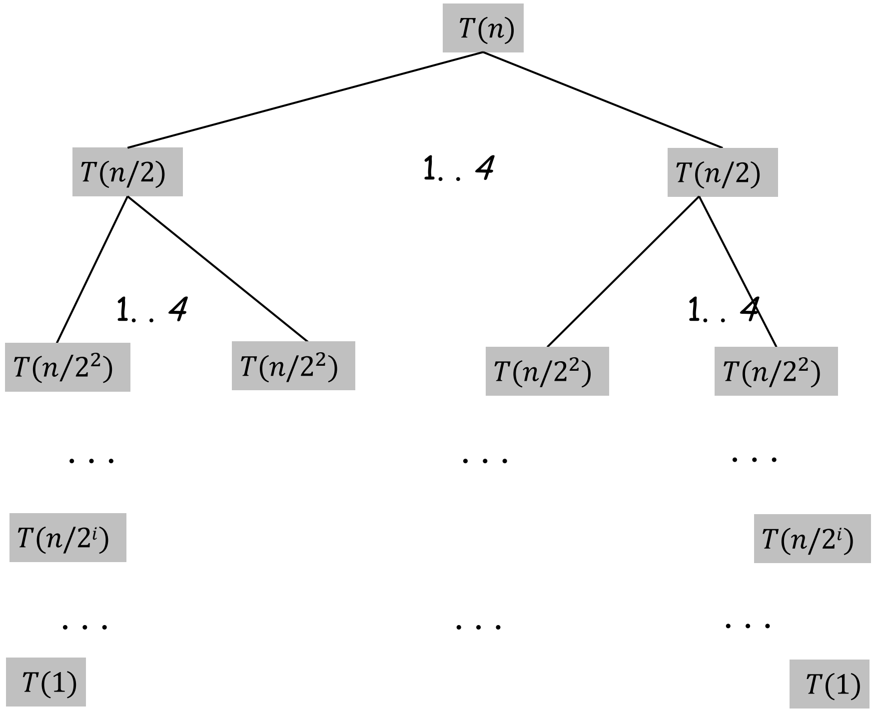
Case 1: $T(n) = 5T\left(\frac{n}{2}\right) + n^2 = \Theta\left(n^{\log_2 5}\right)$



Lv	#pr	work/pr	work/lv
0	1	n^2	n^2
1	5	$(n/2)^2$	$(5/4)n^2$
2	5^2	$(n/2^2)^2$	$(5/4)^2n^2$
i	5^i	$(n/2^i)^2$	$(5/4)^in^2$
$\log_2 n$	$5^{\log_2 n} = n^{\log_2 5}$	1	$n^{\log_2 5}$

$$T(n) = n^{\log_2 5} + n^2 \sum_{j=0}^{\log_2 n - 1} \left(\frac{5}{4}\right)^j$$

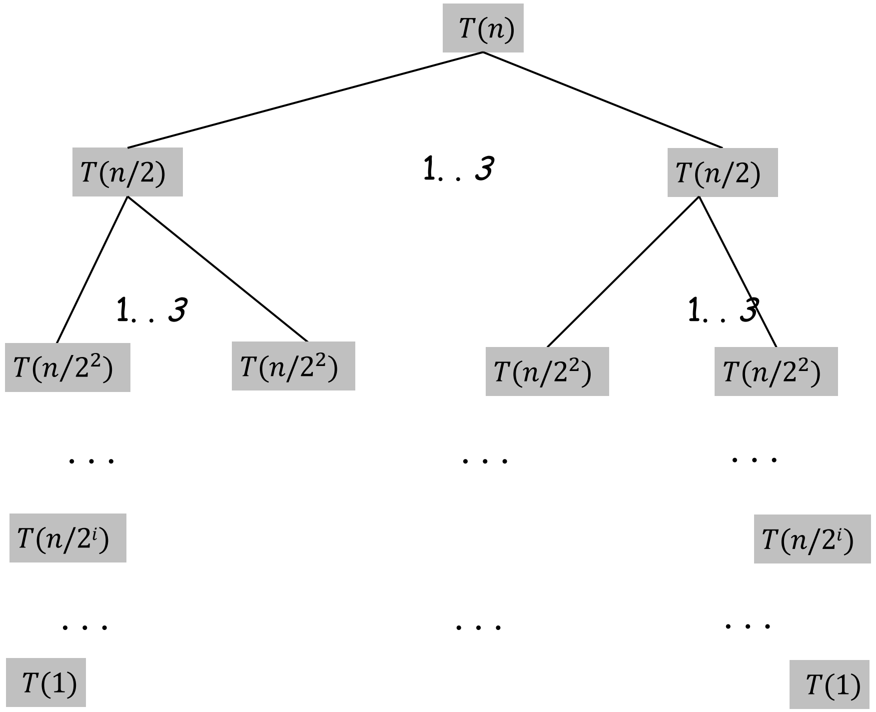
Case 2: $T(n) = 4T\left(\frac{n}{2}\right) + n^2 = \Theta(n^2 \log n)$



Lv	#pr	work/pr	work/lv
0	1	n^2	n^2
1	4	$(n/2)^2$	n^2
2	4^2	$(n/2^2)^2$	n^2
i	4^i	$(n/2^i)^2$	n^2
$\log_2 n$	$4^{\log_2 n} = n^{\log_2 4}$ $= n^2$	1	n^2

$$T(n) = n^2 + n^2 \sum_{j=0}^{\log_2 n - 1} 1$$

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



Lv	#pr	work/pr	work/lv
0	1	n^2	n^2
1	3	$(n/2)^2$	$(3/4)n^2$
2	3^2	$(n/2^2)^2$	$(3/4)^2 n^2$
i	3^i	$(n/2^i)^2$	$(3/4)^i n^2$
$\log_2 n$	$3^{\log_2 n} = n^{\log_2 3}$	1	$n^{\log_2 3}$

$$T(n) = n^{\log_2 3} + n^2 \sum_{j=0}^{\log_2 n - 1} \left(\frac{3}{4}\right)^j$$

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 $f(n) = \Omega(n^{c+\epsilon})$ for some $\epsilon > 0$ => $\mathrm{T}(n) = O(f(n))$

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

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

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 $T(n) = \Theta\left(n^{\log_2 3}\right) = \Theta(n^{1.58...})$

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

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

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

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

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

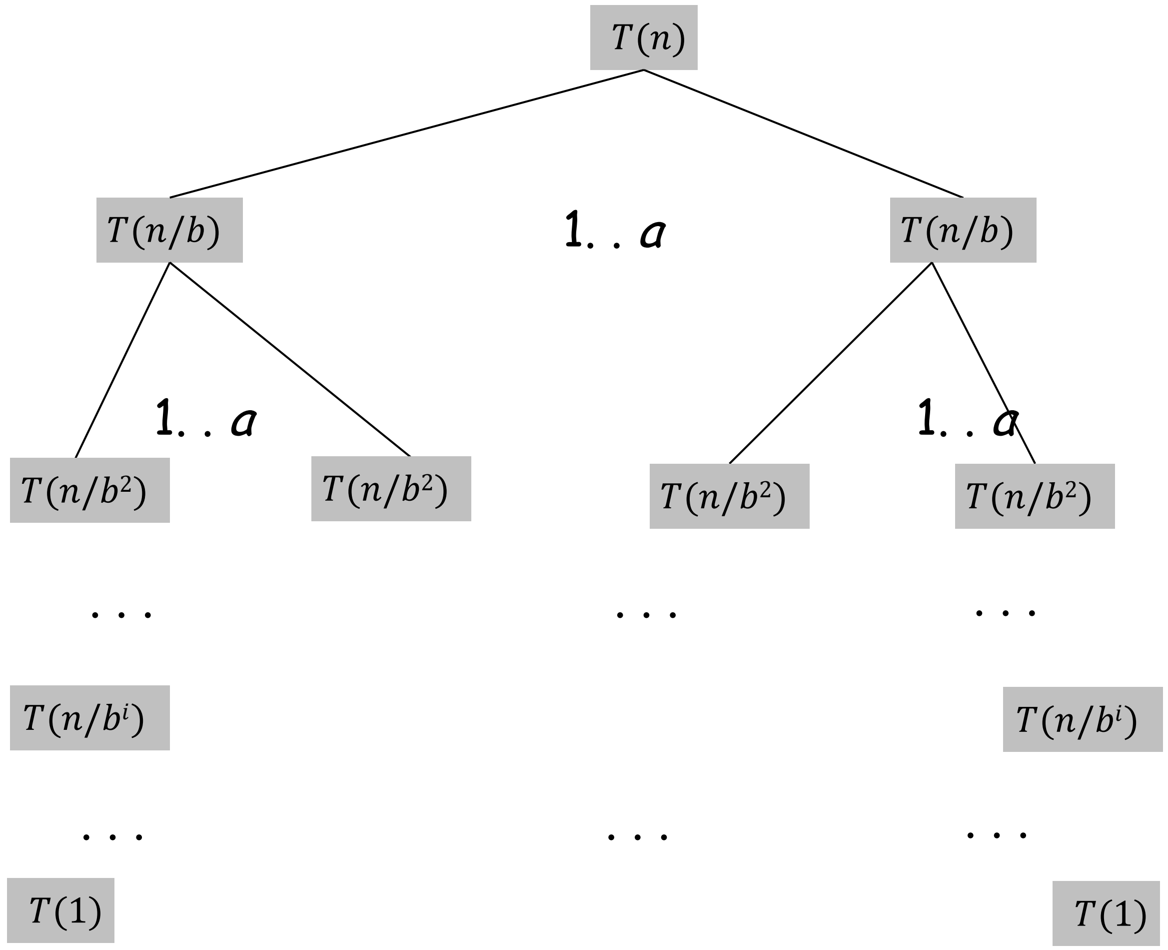
2. If $\mathrm{c} = 1,$ then $T(n) = O(n \log n)$

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, $c = \log_b a$



Lv	#pr	work/pr	work/lv
0	1	kn	kn
1	a	kn/b	$kn(a/b)$
2	a^2	kn/b^2	$kn(a/b)^2$
i	a^i	kn/b^i	$kn(a/b)^i$
$\log_b n$	$a^{\log_b n} = n^{\log_b a} = n^c$	1	n^c

- The total running time is n^c for the bottom level, plus $kn\left(1 + (a/b) + (a/b)^2 + \cdots + (a/b)^{\log_b n - 1}\right)$ for the rest

$\boxed{c = \log_b a}$

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

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

If $a > b, increasing\ geometric\ series$

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

Hence, for $a > b \quad (c > 1)$

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

Bottom level dominates cost.

Example: If $T(n) \le 3T\left(\frac{n}{2}\right) + n$, $a = 3$, $b = 2$, and $T(n) = O\left(n^{\log_2 3}\right) = O(n^{1.58\ldots})$

$c = \log_b a$

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

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

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

Hence, for $a = b \quad (c = 1)$

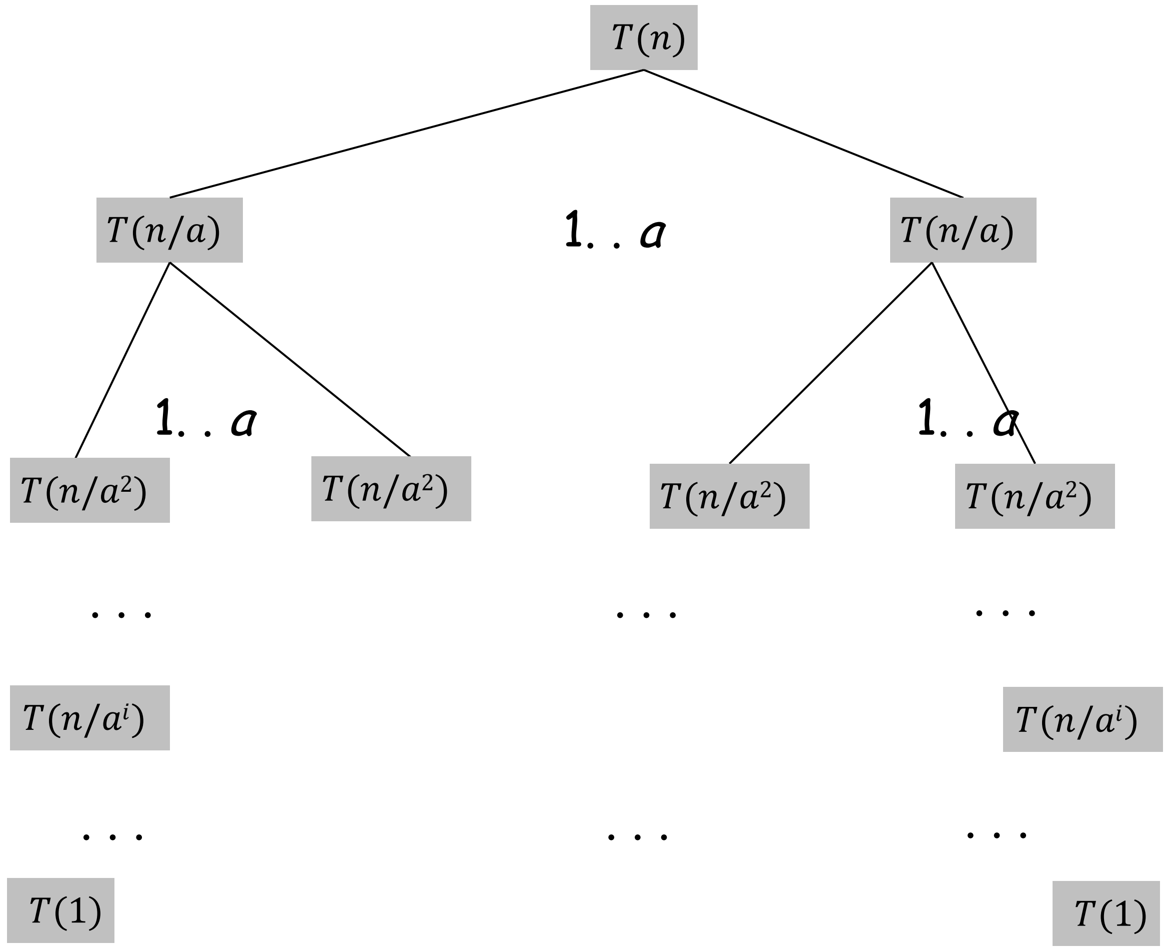
$$\boxed{T(n) = O(n + kn \log_b n) = O(n \log n)}$$

Each level contributes cost kn.

Example: If $T(n) \le 2T\left(\frac{n}{2}\right) + n$, $a = 2$, $b = 2$, and $T(n) = O(n \log n)$

Visualization for Case 2 ($a = b$)

$T(n) \le aT(n/a) + kn,\ c = 1$



Lv	#pr	work/pr	work/lv
0	1	kn	kn
1	a	kn/a	kn
2	a^2	kn/a^2	kn
i	a^i	kn/a^i	kn
$\log_a n$	$a^{\log_a n} = n$	1	n

$\boxed{c = \log_b a}$

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

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

If $a < b$, *decreasing geometric series*

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

Hence, for $a < b \quad (c < 1)$

$$\boxed{T(n) = O(n^c + kn) = O(n)}$$

Top level (0) dominates the cost.

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