# Iteration/Recursion tree method

**A recursion tree can be used to visualize the iteration procedure.**

**Example:**

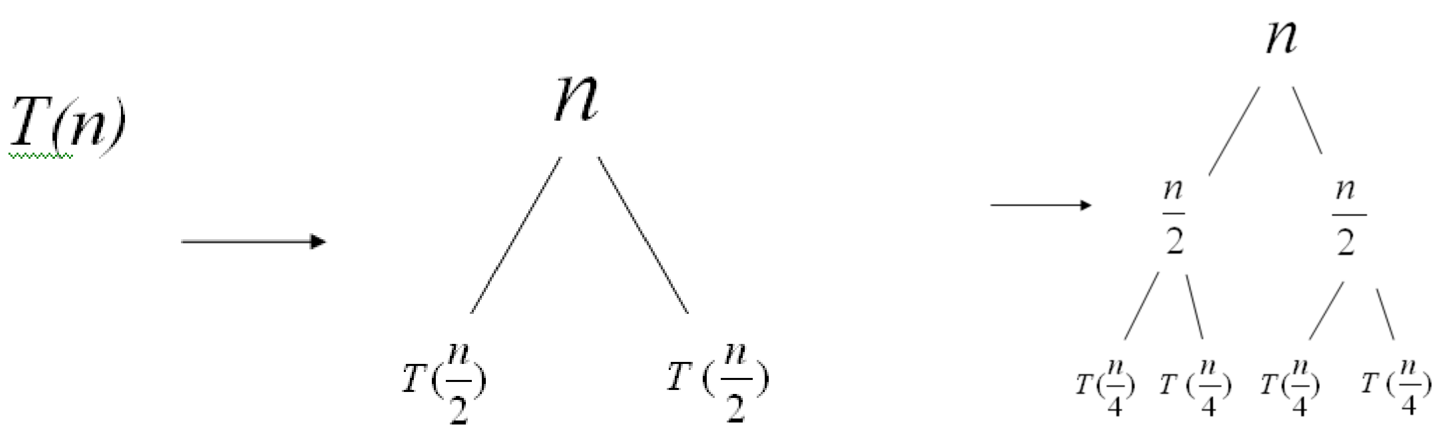
$$T(n) = 2T(n/2) + n$$

$$\begin{aligned}
T(n) &= 2T(\tfrac{n}{2}) + n \\
&= 2(2T(\tfrac{n}{4}) + \tfrac{n}{2}) + n \\
&= 2^2T(\tfrac{n}{2^2}) + n + n \\
&= 2^2(2T(\tfrac{n}{2^3}) + \tfrac{n}{2^2}) + 2n \\
&= 2^3T(\tfrac{n}{2^3}) + 3n \\
&\ldots \\
&= 2^iT(\tfrac{n}{2^i}) + in
\end{aligned}$$

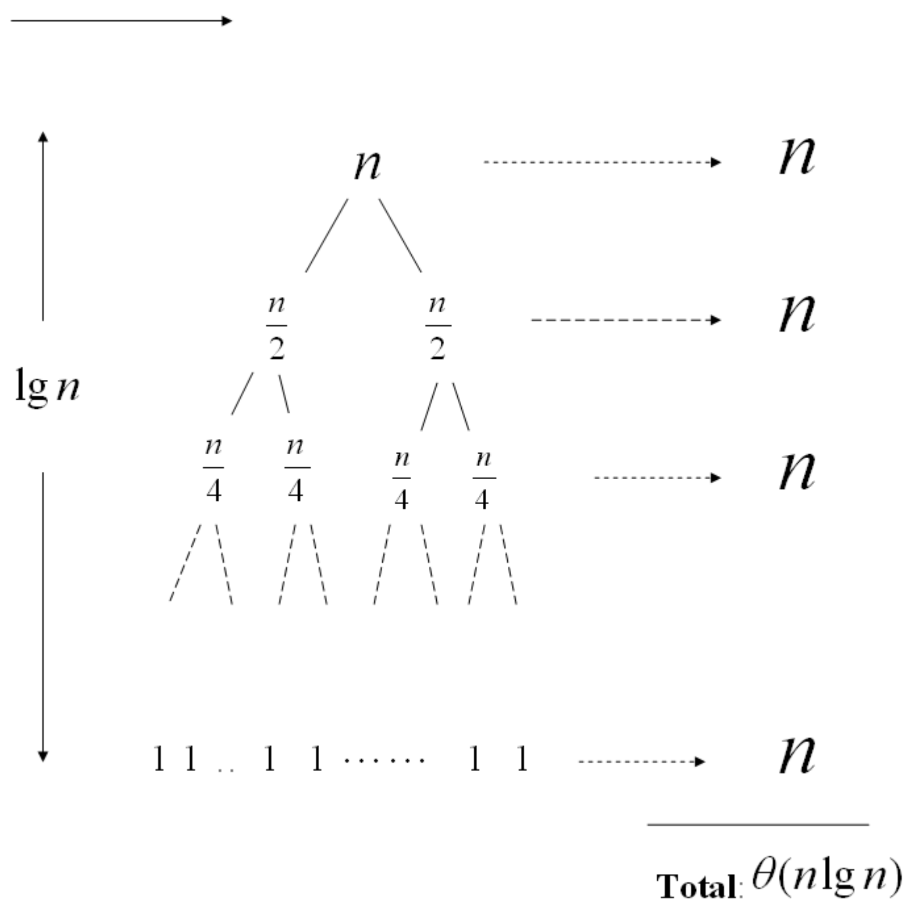
# Using a Recursion Tree

The idea of a Recursion Tree is to expand $T(n)$ to a tree with the same total cost. Two things are important:

- The height of the tree.
- The cost of the nodes at each level.



# Cont'd



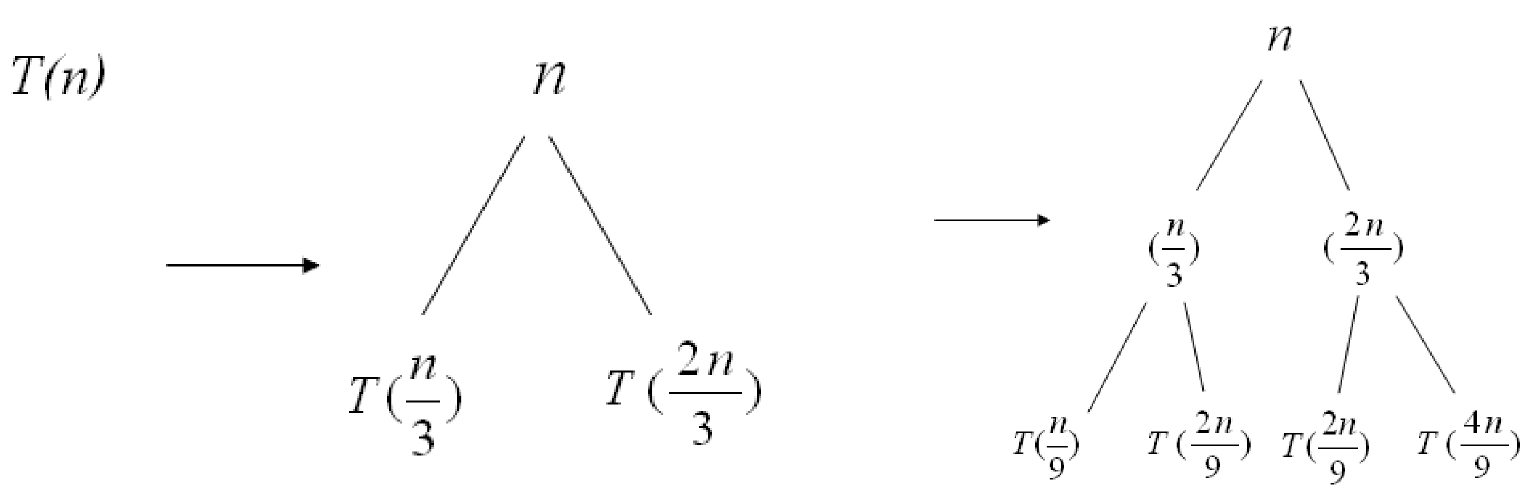
$n$
$\frac{n}{2}$ $\frac{n}{2}$
$\lg n$
$\frac{n}{4}$ $\frac{n}{4}$ $\frac{n}{4}$ $\frac{n}{4}$
1 1 .. 1 1 ...... 1 1
$n$
$n$
$n$
$n$
Total: $\theta(n \lg n)$

# Recursion Tree method, Cont'd

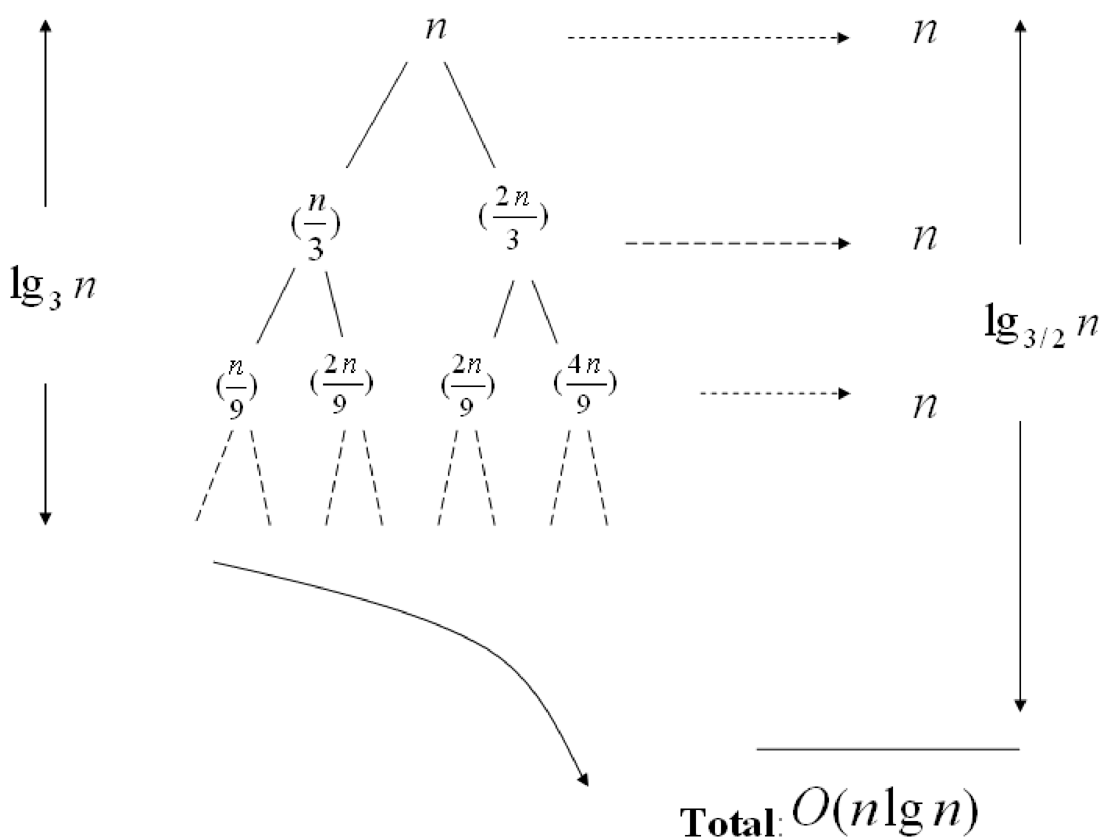
**Using a Recursion Tree is a good way to guess a solution.**

**Example:**

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



# Cont'd


$n$
$(\frac{n}{3})$
$(\frac{2n}{3})$
$(\frac{n}{9})$
$(\frac{2n}{9})$
$(\frac{2n}{9})$
$(\frac{4n}{9})$
$n$
$n$
$n$
$\lg_3 n$
$\lg_{3/2} n$
Total: $O(n \lg n)$

# A more powerful approach: Master Method

## Theorem 4.1 (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

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

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

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 $T(n) = \Theta(n^{log_b a}\ lgn)$.

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

# What does the Master Theorem say?

$f(n)$ vs. $n^{log_b a}$

case 1: $n^{log_b a} \quad > \quad f(n)$, $T(n) = \Theta(n^{log_b a})$.

case 2: $n^{log_b a} \quad = \quad f(n)$, $T(n) = \Theta(n^{log_b a}\ lgn) = \Theta(f(n)\ lgn)$.

case 3: $f(n) \quad > \quad n^{log_b a}$, $T(n) = \Theta(f(n))$.

## Examples

### Example 1

$$\begin{aligned} T(n) &= T(n/5)+1 \\ n^{log_b a} &= n^{log_5 1} = 1 \\ f(n) &= 1 \end{aligned}$$

$$n^{log_b a} \leftrightarrow f(n) : \text{case } 2 \Rightarrow T(n) = \Theta(lgn)$$

### Example 2

$$\begin{aligned} T(n) &= 2T(n/2)+n \\ n^{log_b a} &= n \\ f(n) &= n \end{aligned}$$

$$n^{log_b a} \leftrightarrow f(n) : \text{case } 2 \Rightarrow T(n) = \Theta(nlgn)$$

## Examples

**Example 3**

$$
\begin{aligned}
T(n) &= 3T(n/4) + n^2 \\
n^{log_b a} &= n^{log_4 3} < n^1 \\
f(n) &= n^2
\end{aligned}
$$

$f(n) = \Omega(n^{log_b a+\epsilon})$, where $\epsilon$ can be 0.5. Possibly case 3. Need to check the regular condition: $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$:

$$
\begin{aligned}
af(n/b) &\leq cf(n) \\
3(n/4)^2 &\leq cn^2 \quad \text{as long as} \quad c \geq \frac{3}{16}
\end{aligned}
$$

So case 3: $\Rightarrow T(n) = \Theta(n^2)$

# Examples

**Example 4**

$$\begin{aligned} T(n) &= 9T(n/3) + n \\ n^{log_b a} &= n^{log_3 9} = n^2 \\ f(n) &= n = O(n^{2-\epsilon}), \quad \text{where } \epsilon \text{ can be} < 1. \end{aligned}$$

$$\text{Case 1} \Rightarrow T(n) = \Theta(n^2)$$

**Example 5**

$$\begin{aligned} T(n) &= T(2n/3) + 1 \\ n^{log_b a} &= n^{log_{3/2} 1} = n^0 = 1 \\ f(n) &= 1 \end{aligned}$$

$$\text{Case 2} \Rightarrow T(n) = \Theta(lgn)$$

# Examples

**Example 6:**

$$
\begin{aligned}
T(n) &= 2T(n/2) + nlgn \\
n^{log_b a} &= n^{log_2 2} = n \\
f(n) &= nlgn
\end{aligned}
$$

**Can we find an $\epsilon$ such that $f(n) = \Omega(n^{1+\epsilon})$?**

Based on the definition of $\Omega$,

$$
\begin{aligned}
f(n) = \Omega(n^{1+\epsilon}) &\Leftrightarrow \lim_{n\to\infty} \frac{n^{1+\epsilon}}{nlgn} = constant \\
&\Leftrightarrow \lim_{n\to\infty} \frac{n^{\epsilon}}{lgn} = constant
\end{aligned}
$$

But $\lim_{n\to\infty} \frac{n^{\epsilon}}{lgn} = \infty \neq constant$, so this example falls into the gap of master method.