Recurrence Equation

Algorithm Analysis

School of CSEE

Exercise 3

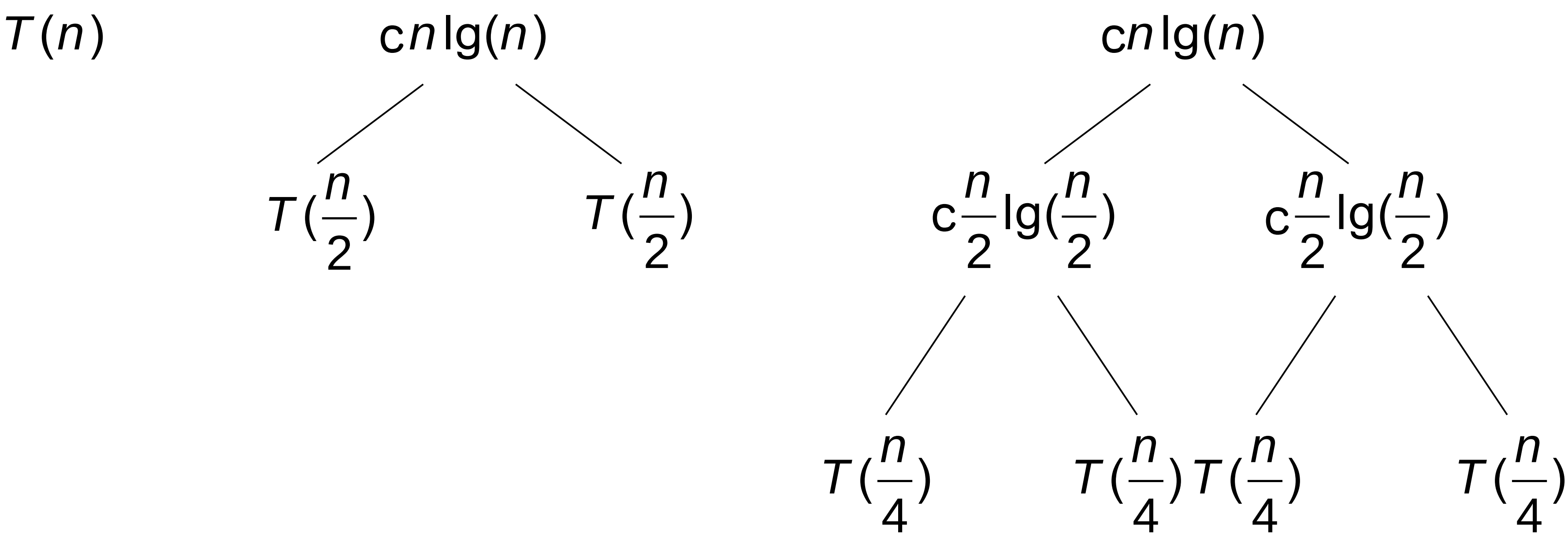
$$T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 2T(n/2) + \Theta(n \lg(n)) & \text{if } n>1 \end{cases}$$

(1) With recursion tree method

(2) With master theorem method

With recursion tree

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



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

$cn\lg(n)$

$c\frac{n}{2}\lg\frac{n}{2}$ $\quad$ $c\frac{n}{2}\lg\frac{n}{2}$

$c\frac{n}{4}\lg\frac{n}{4}$ $\quad$ $c\frac{n}{4}\lg\frac{n}{4}$ $\quad$ $c\frac{n}{4}\lg\frac{n}{4}$ $\quad$ $c\frac{n}{4}\lg\frac{n}{4}$

....

....

....

$T(1)$ $\quad$ $T(1)$ $\quad$ $T(1)$ $\quad$ $T(1)$

$\lg n$

$cn\lg(n)$

$cn\lg(\frac{n}{2})$

$cn\lg(\frac{n}{4})$

$\Theta(2^{\log_2 n})$

With recursion tree

$$T(n) = cn\lg n + cn\lg(n/2) + \ldots cn\lg(n/2^{(\lg n-1)}) + \Theta(2^{\log_2 n})$$

$$= cn*\{(\lg n + \lg n + \ldots) – (0+1+2\ldots +(\lg n-1)\} + \Theta(2^{\log_2 n})$$

$$= cn*\{\lg^2 n – ½ \lg n(\lg n-1)\} + \Theta(n)$$

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

With recursion tree

$T(n) = cn\lg n + cn\lg(n/2) + \ldots\ cn\lg(n/2^{(\lg n-1)}) + cn\lg(n/2^{(\lg n)})$

$= cn*\{(\lg n + \lg n + \ldots) - (0+1+2\ldots +(\lg n-1)+\lg n\ \}$

$= cn*\{\lg n*(\lg n +1) - ½\ \lg n*(\lg n+1)\}$

$= cn*\{½*\lg n*(\lg n +1)\}$

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

With master theorem

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

- a = 2, b = 2. Thus $n^{\log_b a} = n$
- $f(n) = n\lg(n)$
- $f(n) = O(n^{\log_b a - \varepsilon})$ X
- $f(n) = \Theta(n^{\log_b a})$ X
- $f(n) = \Omega(n^{\log_b a + \varepsilon})$ X

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} \to c > 0 \quad \Rightarrow \quad f(n) = \Omega(g(n))$$

Exercise 4

Solve the following recurrence equation with master theorem.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 4T(n/4) + \lg(n) & \text{if } n>1 \end{cases}$$

With master theorem

- a = 4, b = 4. Thus $n^{\log_b a} = n$
- $f(n) = \lg(n)$
- Case1: $f(n) = O(n^{\log_b a-\varepsilon})$?

$\lg(n) = O(n^{1-\varepsilon})$ for $\varepsilon = ½$, *etc*

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

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \to c < \infty \quad \Rightarrow \quad f(n) = O(g(n))$$

Exercise 5

- Prove $1 + 3 + 5\ldots + (2n-1) = n^2$
  - Base case:
    - When $n = 1$, then L.H.S =1 and R.H.S. = 1^2 =1
  - Inductive hypothesis:
    - For n greater than 0, assume that $1 + 3 + 5\ldots + (2k-1) = k^2$ holds true all $k \geq 0$ such that $k < n$.
    - By hypothesis the formula is true when $k = n-1$,
      $1 + 3 + 5\ldots + (2n-3) = (n-1)^2$
  - Proof of goal statement:
    $1 + 3 + 5\ldots + (2n-3) + (2n-1) = (n-1)^2 + (2n-1) = n^2$

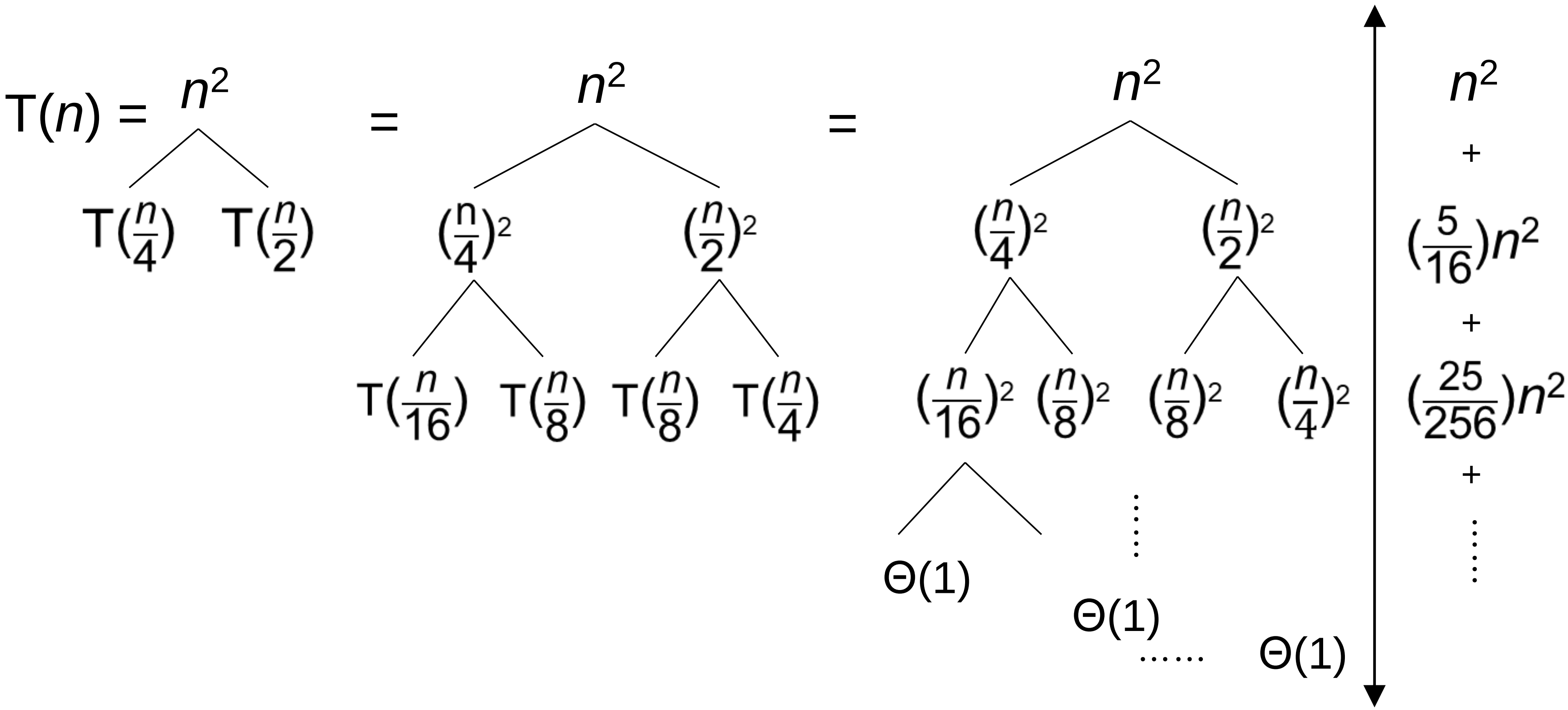
Exercise 6

Using the recursion tree method, prove that **T(n) = $\Theta(n^2)$.**

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

Answer 6

$$T(n) = T(\tfrac{n}{4}) + T(\tfrac{n}{2}) + n^2$$



Level by level total:

$$T(n) \le n^2\left(1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \ldots + \left(\frac{5}{16}\right)^k + \ldots\right)$$

$$\le n^2\left(1/\left(1 - \left(\frac{5}{16}\right)\right)\right) < 2n^2$$

Thus, $T(n) = O(n^2)$

And, since $T(n) \ge n^2$ (from recurrence equation)

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

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