Algorithm Design and Analysis

วิชาบังคับก่อน: 204251 และ 206281
ผู้สอน: ตอน 1 ผศ. เบญจมาศ ปัญญางาม เรียน ห้อง 100
ตอน 2 ผศ. ดร. จักริน ชวชาติ เรียน ห้อง 209

บทที่ 4
การแก้ปัญหาความสัมพันธ์แบบรีเคอร์เรนซ์
(Solving recurrence relations)
Part II

Recursion-tree Method

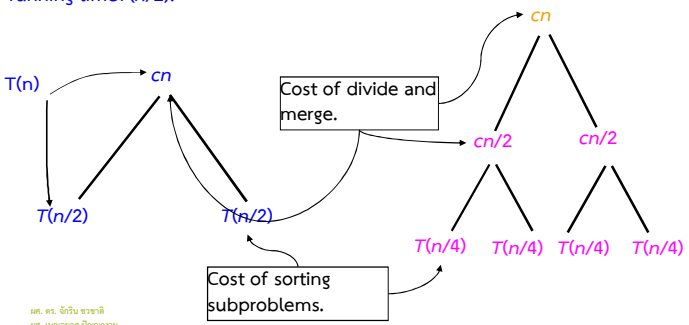
- Making a good guess is sometimes difficult with the substitution method.
- Use recursion trees to devise good guesses.
- Recursion Trees
  - Show successive expansions of recurrences using trees.
  - Keep track of the time spent on the subproblems of a divide and conquer algorithm.
  - Help organize the algebraic bookkeeping necessary to solve a recurrence.

Recursion-tree Method

Example: Merge sort analysis ➔ T(n) =2T(n/2)+cn

For the original problem, we have a cost of *cn*, plus two subproblems each of size (*n*/2) and running time *T*(*n*/2).

Each of the size *n*/2 problems has a cost of *cn*/2 plus two subproblems, each costing *T*(*n*/4).

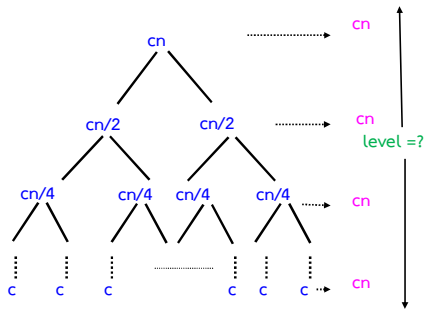


Recursion-tree Method

Example: Merge sort analysis ➔ T(n) =2T(n/2)+cn, T(1)= c

- Continue expanding until the problem size reduces to 1.
- Each time we go down one level,
  - number of sub-problems doubles (2, 4, 8 ..., 2^k)

➔ จำนวนโหนดในแต่ละ level
- but the cost per sub-problem halves (cn/2, cn/4, cn/8,....)

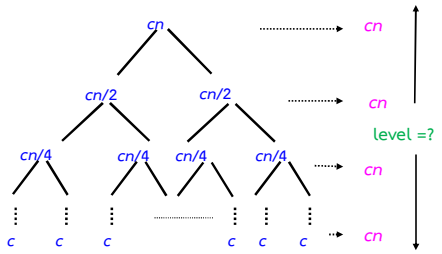


Total T(n) = ?

- Each level has total cost cn ➔ cost per level remains the same.

Recursion-tree Method

Example: Merge sort analysis ➔ T(n) =2T(n/2)+cn, T(1)= c



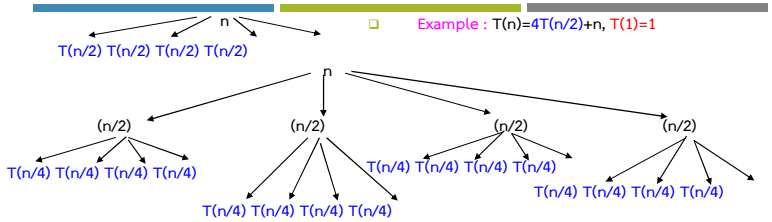
- Each level has total cost cn
- There are log *n* +1 levels, k = log *n*
- Total cost = sum of costs at each level
  = (log *n* +1) *cn* = *c n*log *n* + *cn*
  = Θ(*n* log *n*).

Total T(n) = *n* log *n*

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

Recursion-tree Method

❑ Example : T(n)=4T(n/2)+n, T(1)=1



∴ T(n) =O(n^2)

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

Recursion-tree Method

Compare to Iterative substitution method

Example : T(n)=4T(n/2)+n ,T(1)=1

=4[4T(n/4)+n/2]+n
=4[4T(n/2^2)+n/2)]+n
=4^2T(n/2^2)+2n+n

k=1

∴ T(n) =O(n^2)

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

The Master Method

- ❑ Based on the Master theorem.
- ❑ "Cookbook" approach for solving recurrences of the form

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

  - a ≥ 1, b > 1 are constants.
  - f(n) is asymptotically positive.
  - Requires memorization of three cases.

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

The Master Method

- Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function,
- Let $T(n)$ be defined on nonnegative integers by the recurrence
- $T(n) = aT(n/b) + f(n)$, where we can replace n/b by $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$.
- $T(n)$ can be bounded asymptotically in three cases:

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

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

The Master Method

- Example

1. $T(n) = 16T(n/4)+n$
   - $a = 16$, $b = 4$, $f(n) = n$
   - $n^{\log_b a} = n^{\log_4 16} = n^2$.
   - Is $f(n) = n = O(n^{\log_b a - \varepsilon}) = O(n^{2-\varepsilon})$, $\varepsilon > 0$?
   - Why ?
   - Yes, where $\varepsilon = 1 \Rightarrow$ Case 1.
   - Hence, $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$.
2. $T(n) = T(3n/7) + 1$
   - $a = 1$, $b=7/3$, $f(n) = 1$
   - $n^{\log_b a} = n^{\log_{(7/3)} 1} = n^0 = 1$
   - Is $f(n) = 1 = O(n^{\log_b a - \varepsilon}) = O(n^{0-\varepsilon})$, $\varepsilon > 0$? ➔ No
   - Why ?
   - Next, Is $f(n) = 1 = \Theta(n^{\log_b a})$? ➔ Yes $\Rightarrow$ Case 2.
   - Therefore, $T(n) = \Theta(n^{\log_b a} \log n) = \Theta(\log n)$

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

The Master Method

- Example

3. $T(n) = 3T(n/4) + n \log n$
   - $a = 3$, $b=4$, thus $n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$
   - Is $f(n) = n \log n = O(n^{\log_4 3 - \varepsilon})$, $\varepsilon > 0$? ➔ No
   - but, $f(n) = n \log n = \Omega(n^{\log_4 3 + \varepsilon})$ where $\varepsilon \approx 0.2 \Rightarrow$ Case 3.
   - and for sufficiently large n,

     a $f(n/b) = 3(n/4 \log n/4) \leq c\,(n \log n)$

     $= cf(n)$ is true where $c = 3/4 < 1$
   - Therefore, $T(n) = \Theta(f(n)) = \Theta(n \log n)$.

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม

The Master Method

- Example

4. $T(n) = 2T(n/2) + n \log n$
   - $a = 2$, $b=2$, $f(n) = n \log n$, and $n^{\log_b a} = n^{\log_2 2} = n$
   - $f(n)$ is asymptotically larger than $n^{\log_b a}$, but not polynomially larger. The ratio $\log n$ is asymptotically less than n^{ε} for any positive ε. Thus, the Master Theorem *doesn't* apply here.

ผศ. ดร. จักริน ชวชาติ
ผศ. เบญจมาศ ปัญญางาม