for(1 ~ n) . 두 배열에 값 넣기

function call (좌)

function call (우)

Algorithm Analysis Homework 2

Due by 3/31(Fri.) through LMS

1. Answer the following questions for Quicksort algorithm you have learned in data structure course.

   (a) Determine the recurrence equation (T(n)) for best case (best case: array is divided into two equal sized sub array).

   (b) Solve above equation with master theorem method. Express time complexity in 'Theta' notation.

   (c) You are goint to check the answer you derived in part (b) with recursion tree method. Draw the recursion tree in your exercise note and answer the following questions.

   (i) What is the height of the tree?

   (ii) Determine the number of nodes at level 2. ('root' is at level 0.)

   (iii) Determine level sum of level 1 and 2. (level1, level2 각각 구하시오)

   (iv) Determine asymptotic **tight bound** solution from the tree. Is your answer same as the one from part (b)

2. Use the master theorem method to give tight asymptotic bounds for the following recurrences.

   (a) $T(n) = 9T(n/3) + \Theta(n^2)$

   (b) $T(n) = 3T(n/3) + \Theta(n^2 \log n)$

1. (a) Determine the recurrence equation (T(n)) for best case (best case: array is divided into two equal sized sub array).

$$T(n) \begin{cases} \theta(1) & (n=1) \\ 2T(\frac{n}{2})+n & (n>1) \end{cases}$$

1. (b) Solve above equation with master theorem method. Express time complexity in 'Theta' notation.

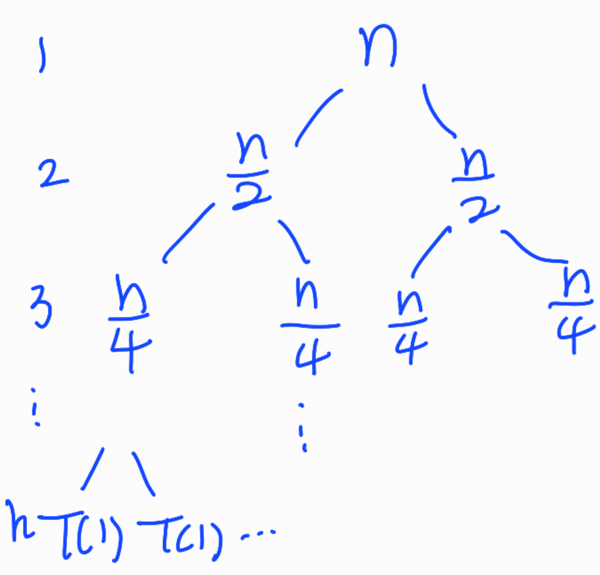
$a=2,\ b=2,\ f(n)=n.$

$$n = O(n^{\log_2 2}) = O(n)$$

case 2 of master theorem.

$$\boxed{\therefore T(n) = \theta(n \lg n)}$$

(c) You are goint to check the answer you derived in part (b) with recursion tree method. Draw the recursion tree in your exercise note and answer the following questions.



각 항 x 항의수 = sum

$(\frac{1}{2})^0 n \times 2^0 = n$

$(\frac{1}{2})^1 n \times 2^1 = n$

$(\frac{1}{2})^2 n \times 2^2 = n$

$\vdots$

$\underbrace{(\frac{1}{2})^h n}_{=1} \times \underbrace{2^h}_{=2^{\lg n}=n} = n$

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

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

$2T(\frac{n}{8})+\frac{n}{4}\ \ldots$

$\hookrightarrow (\frac{1}{2})^h n = 1.$

$n = 2^h$

$\boxed{\therefore h = \log_2 n}$

(i) What is the height of the tree?

$h = \boxed{\lg n}$

(ii) Determine the number of nodes at level 2. ('root' is at level 0.)

$2^2 = \boxed{4}$

(iii) Determine level sum of level 1 and 2. (level1, level2 각각 구하시오)

$\boxed{\text{lv}1 : n, \ \text{lv}2 : n}$

(iv) Determine asymptotic tight bound solution from the tree. Is your answer same as the one from part (b)

$T(n) = n \times \lg n = n \lg n, \ \boxed{\Theta(n \lg n)}$

2. Use the master theorem method to give tight asymptotic bounds for the following recurrences.

(a) T(n) = 9T(n/3) + Θ(n^2)

$a=9, \ b=3, \ f(n) = n^2$

$n^2 = \Theta(n^{\log_3 9}) = \Theta(n^2), \ \text{case}2, \ \boxed{T(n) = \Theta(n^2 \lg n)}$

(b) T(n) = 3T(n/3) + Θ(n^2logn)

$a=3, \ b=3, \ f(n) = n^2 \lg n$

$n^2 \lg n = \Omega(n^{\log_3 3 + \varepsilon}) = \Omega(n^{1+\varepsilon}), \ \text{and..}$

$\Rightarrow \text{case}3, \ \boxed{T(n) = \Theta(n^2 \lg n)}$

$3 \cdot (\frac{n}{3})^2 \lg \frac{n}{3} \le c \cdot n^2 \lg n$

$\rightarrow \frac{1}{3} n^2 \lg \frac{n}{3} \le c \cdot n^2 \lg n$

$\rightarrow \frac{1}{3} n^2 (\lg n - \lg 3) \le c n^2 (\lg n)$

✡ $c = \frac{1}{3}$일 때 성립함 ✡

$\boxed{\text{Case 3의 조건 2}}$ $a f(\frac{n}{b}) \leq c \cdot f(n)$

$\boxed{c < 1}$

끝