Algorithm Analysis(01)

21700034 곽영혜

1. Please answer if it is true or false for each equation. (1 point/each)

a. $3n^2+1 = O(n^3)$

b. $n^3/10 + n\log_2 n = O(n^3)$

c. $n^2+n = \Omega(n^3)$

d. $n^{3/2}+23n^2 = \Theta(n^{3/2})$

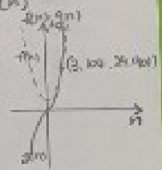
e. $n\log n = o(n\log n)$

f. $n^{3/2}+10^2n^2 = \omega(n)$

a) $O(g(n)) = f(n)$ (Big Oh)

there exist positive constants C & n_0 such that $0 \le f(n) \le Cg(n)$ for all $n \ge n_0$

a. $f(n) = 3n^2+1$, $g(n) = n^3$, $C = 1$, $n_0 \doteq 3.104$



If f(n) expressed in polynomial, leading term's highest degree will be Big Oh. $\Rightarrow 3n^2+1 \Rightarrow O(n^2)$

a = True $\quad 3n^2+1 = O(n^2)$

b) Big Oh

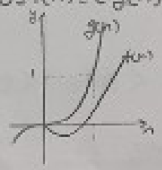
$f(n) = n^3/10 + n\log_2 n$

$g(n) = n^3$

$C = 0$, $n_0 = 0$

positive C & n_0

$0 \le f(n) \le C \cdot g(n)$



b = True

c) Big Omega

$\Omega(g(n)) = f(n)$

positive constants C & n_0

$0 \le Cg(n) \le f(n)$, $n \ge n_0$

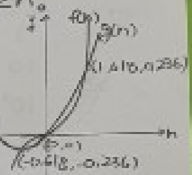
c. $f(n) = n^2+n$

$g(n) = n^3$

$C = 1$

$-\infty \sim -0.618$ & $0 \sim 1.618$

(n_0 doesn't exist)



c = False

d) Big Theta $\longrightarrow$ Big Oh & Omega

$0 \le f(n) \le C g(n)$ $\quad 0 \le Cg(n) \le f(n)$

$\Theta(g(n)) = f(n)$

$f(n) = n^{3/2}+23n^2$

$g(n) = n^{3/2}$

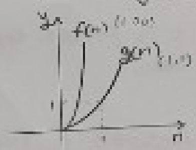
$C = 0$, $n_0 = 0$

$C = 24$, $n_0 = 1$

$\Rightarrow f(1) = C \cdot g(1)$

$24 = 24g(1)$

$= 24 \cdot 1$



Big Oh → X

Big Omega → O

d = False

e) Little Oh

$o(g(n)) = f(n)$ $\quad 0 \le f(n) < Cg(n)$, $n \ge n_0$

$f(n) = n\log n$

$g(n) = n\log n$

$\Rightarrow$ False

ex) $n-5, n, n^2, 10^{10}n^2, n^2-9 \in O(n^3)$

$n^3 \notin O(n^3)$

$\Rightarrow n\log n \notin O(n\log n)$

e = False

f) Little Omega

$\omega(g(n)) = f(n)$

$C > 0$, $n_0 > 0$

$0 \le Cg(n) < f(n)$

$n \ge n_0$

$f(n) = n^{3/2}+10^2n^2$

$g(n) = n$

ex) $n^2, 10^{9}n^3+10^3, n^3-9 \in \omega(n)$

$n^{3/2}+10^2n^2 \in \omega(n)$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n^{3/2}+10^2n^2}{n} = \lim_{n\to\infty}\frac{n^{\frac{1}{2}}+10^2n}{1} \Rightarrow \infty$$

$\Rightarrow f(n) = \omega(g(n))$

f = True

2. Arrange the following functions in ascending order of growth rate (3 points)

a. $n, n^2, n^3, 2^n, \log n, n\log n, n2^n, \sqrt{n}$

n	n^2	n^3	2^n	$\log n$	$n\log n$	$n2^n$	$\sqrt{n}$
10	10^2	10^3	2^{10}	1	10	$10\cdot2^{10}$	$\sqrt{10}=3.16\ldots$
10^2	10^4	10^6	2^{10^2}	2	$2\cdot10^2$	$10^2\cdot2^{10^2}$	10
10^3	10^6	10^9	2^{10^3}	3	$3\cdot10^3$	$10^3\cdot2^{10^3}$	$10\sqrt{10}$
			$= 1.071509e+301$				

Answer: $\log n, \sqrt{n}, n, n\log n, n^2, n^3, 2^n, n2^n$

3. Arrange the following functions in ascending order of growth rate

a. $n\log n, e^n, n!, n^{1/100}$

e^n= n개의 e의 곱
n! = n

n	$n\log n$	e^n	$n!$	$n^{1/100}$
1	0	e	1	1
10	10	e^{10}	$10!$	$10^{0.01}$
10^2	$2\cdot10^2$	e^{10^2}	$100!$	$10^{0.02}$
	$= 200$	$= 2.6881171e+43$	$= 9.33262e+157$	$= 1.0471285\ldots$

$n^{1/100} < n\log n < e^n < n!$

Answer: $n^{1/100}, n\log n, e^n, n!$

4. Which of following algorithms is optimal if it is proven that the given problem needs at least $n\log n$ basic operations?

a. Algorithm 1: $\theta(n\log 2^n)$

b. Algorithm 2: $\theta(n^2)$

c. Algorithm 3: $\theta(n\log n^2)$

Answer: C. Algorithm 3 : $\theta(n\log n^2)$

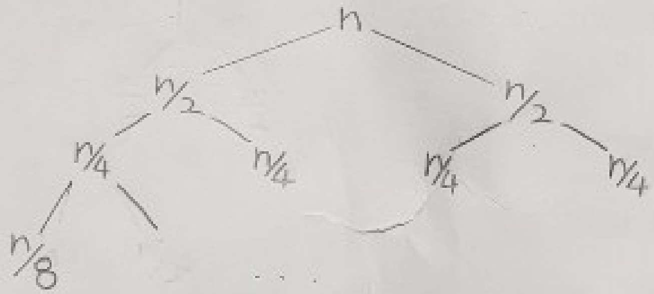
a. $\theta(n\log 2^n) = \theta(n^2\log 2)$

c. $\theta(n\log n^2) = \theta(2n\log n)$

Try to solve a problem at minimum work, C. Algorithm 3 is optimal. $\theta(2\times n\log n)$

5. Use the recursion-tree method to determine a good asymptotic upper bound on the recurrence.

a. $T(n) = 2T(\frac{n}{2}) + n$, where $T(1) = 1$



level(i)	Num
0	$1 = 2^0$
1	$2 = 2^1$
2	$4 = 2^2$
3	$8 = 2^3$

$$T(n) = n + 2(n/2) + 4(n/4) + 8(n/8) + \cdots$$
$$= n + n + n + n + \cdots$$
$$= n(1 + 1 + 1 + 1 + \cdots)$$
$$= n \cdot n = n^2 \Longrightarrow \underline{T(n) = O(n^2)}$$