1. By Dangnhi Ngo

1/ Function Order of Growth

Smallest $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ Largest

(1) n^{-3} $\qquad$ (2) $lg(2^{lg(n^3)})$ $\qquad$ (3) $(lgn)^3$ $\qquad$ (4) n^3

a/ Compare $n^{-3} < lg(2^{lg(n^3)})$

$lg(2^{lg(n^3)}) = lg(n^3)$

$n^{-3} = O(lg(n^3)) \Rightarrow 0 \leqslant \frac{1}{n^3} \leqslant c.\, lg(n^3)$

Let $c = 1$, $\quad 0 \leqslant \frac{1}{n^3} \leqslant lg(n^3)$

$0 \leqslant 1 \leqslant n^3 lg(n^3)$

$n_0 = 2$

b/ Compare $lg(2^{lg(n^3)}) < (lg\,n)^3$

$lg(2^{lg(n^3)}) = lg(n^3) = O((lg\,n)^3)$

$\Rightarrow \quad 0 \leqslant lg(n^3) \leqslant c.(lgn)^3$

Let $c = 1$, $\quad 0 \leqslant lg(n^3) \leqslant (lgn)^3$

$n_0 = 4$

c/ Compare $(lgn)^3 < n^3$

$(lgn)^3 = O(n^3)$

$\Rightarrow \quad 0 \leqslant (lgn)^3 \leqslant c.n^3$

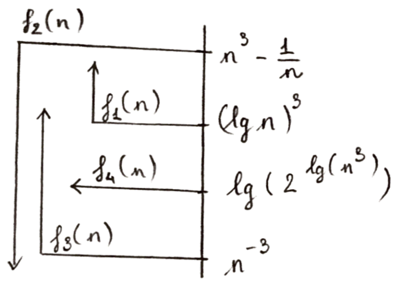
Let $c = 1$, $\quad 0 \leqslant (lgn)^3 \leqslant n^3$

$n_0 = 1$

2. By Dangnhi Ngo

2/ O, Ω, Θ Notation Practice

a/ Arrow diagram



b/ (b) $f_4(n) \in O(f_1(n))$

TRUE, because there is no value for n where f_1 is not going to be the upper bound for f_4. f_1 is always the upper bound for f_4, and there is no constant value that can be multiplied to f_1 that would not make it an upper bound.

(c) $f_2(n) \in \Omega(f_3(n))$

FALSE, because f_3 is no longer the lower bound for f_2 when $n = 1$

$(f_2 = 0 < f_3 = 1)$

(d) $f_1(n) \in O(f_2(n))$

FALSE, because f_2 will no longer be the upper bound for f_1 when $n = \frac{1}{2}$

$(f_1 > f_2)$

(e) $f_4(n) \in \Theta(\lg^3 n)$

$f_4(n) \in \Theta(f_1(n))$

FALSE, because f_1 is strictly upper bound for f_4, they are not bounds to one another. There is no value of n or c that would make f_1 be lower bound for f_4.

3. By Ben Albert

3)

1. $Max(F(n), g(n)) = \Theta(F(n) + g(n))$

2. $\Theta(F(n)+g(n)) \Rightarrow O(F(n)+g(n))$ & $\Omega(F(n)+g(n))$

3. if $F(n) \geq g(n)$ $Max(F(n), g(n)) = F(n)$
4. if $g(n) > F(n)$ $Max(F(n), g(n)) = g(n)$

5. For $C = 1$, $F(n) \leq F(n)+g(n)$ & $g(n) \leq F(n)+g(n)$

6. therefore, $max(F(n), g(n)) = O(F(n)+g(n))$

7. $Max(F(n), g(n)) \geq F(n)$ & $Max(F(n), g(n)) \geq g(n)$

8. For $C = 1/2$, $Max(F(n), g(n)) \geq (F(n)+g(n)) \cdot C$

9. therefore, $Max(F(n), g(n)) = \Omega(F(n)+g(n))$

10. therefore, $Max(F(n), g(n)) = \Theta(F(n)+g(n))$

4. By Ben Albert

Asymptotically, $n \lg(n)$ is greater than $256n$. However, this is only true for values of n larger than 2^{256}. Therefore, as long as the client is working with less than 2^{256} values. I would reccomend the $F_1(n) = n \lg(n)$ algorithm to them.

$n \lg(n) \geq 256n$

$\lg(n) \geq 256$

$n \geq 2^{256}$

5. By Duyen Tran

Mystery(n)	cost	times
if n is an even number	c_1	1
for i = 1 to n	c_2	$n+1$
for j = n downto n/2	c_3	$(n/2+1) * n \rightarrow n^2/2 + n$
print "1"	c_4	$(n/2) * n \rightarrow n^2/2$
else	c_5	
for k = 1 to n/4	c_6	$n/4+1$
for m = 1 to n	c_7	$(n+1) * n/4$
print "2"	c_8	$n * n/4$

$$T(n) = c_1 + c_2(n+1) + c_3(n^2/2 + n) + c_4(n^2/2)$$

$$c_1 + c_2 n + c_2 + c_3\frac{n^2}{2} + c_3 n + c_4\frac{n^2}{2}$$

$$(c_1 + c_2) + (c_2 + c_3)n + (c_3 + c_4)\frac{n^2}{2}$$

$$a + bn + cn^2 \longrightarrow \Theta(n^2)$$

$$T(n) = c_6\left(\frac{n}{4}+1\right) + c_7\left(\frac{n^2}{4} + \frac{n}{4}\right) + c_8\left(\frac{n^2}{4}\right)$$

$$c_6\frac{n}{4} + c_6 + c_7\frac{n^2}{4} + c_7\frac{n}{4} + c_8\frac{n^2}{4}$$

$$c_6 + (c_6 + c_7)\frac{n}{4} + (c_7 + c_8)\frac{n^2}{4}$$

$$a + bn + cn^2 \rightarrow \Theta(n^2)$$

$$\boxed{T(n) = \Theta(n^2)}$$