1. From Nicholas Quinn.

Nicholas Quinn
202

(1) Slowest

$\lg(2^{\lg(n^3)}) \quad n^2 \lg\lg(n) \quad n^2 \lg(n) \quad (1/3)^{-n} \quad (-3)^{2n}$

1(a) $\lg(2^{\lg(n^3)}) = \log(n^3) = 3\log(n)$

$3\log(n) = O(n^2 \lg\lg(n))$

$0 \le 3\log(n) \le n^2 \lg\lg(n) \cdot C$, Let $C = 1$

$0 \le 3 \le \frac{n^2 \lg\lg(n)}{\log(n)}$

$n_0 = 3$

1(b) $n^2 \lg\lg(n) = O\, n^2 \lg(n)$

$0 \le n^2 \lg\lg(n) \le n^2 \lg(n) \cdot C$, Let $C = 1$

$0 \le \lg\lg(n) \le \lg(n)$

True when $C = 1$ and $n_0 = 3$

1(c) $n^2 \lg\lg(n) = O((1/3)^{-n})$

$0 \le n^2 \lg(n) \le 3^n C$, Let $C = 1$

$0 \le \lg(n) \le \frac{3^n}{n^2}$

$n_0 = 1$

1(d) $1/3^{-n} = O((-3)^{2n})$

$0 \le 3^n \le C 3^{2n}$, let $C = 1$

$n \le 2n$

$1 \le 2$

$n_0 = 1$

2. From Nicholas Bishop.

2 a.

- $f_4(n)$ — $(-3)^{(2n)}$
- $f_3(n)$ — $\left(\frac{1}{3}\right)^{-n}$
- $f_1(n)$ — $n^2 \lg n$
- $f_2(n)$ — $n^2 \lg\lg n$
- $f_5(n)$ — $\lg(2^{\lg(n^3)})$

b. TRUE; $n^2 \lg\lg n \in O(n^2 \lg n)$ $\therefore$

$f_2(n) \in O(n^2 \lg n)$

$f_1(n) \in \Omega(n^2 \lg n)$ $\therefore$

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

c. False, for $f_2(n) = \lg\lg n$ and $f_5(n) = \lg n$, $f_2(n)$ grows asymptotically slower than $f_5(n)$

d. false; for $f_4(n) = (-3)^{(2n)}$ and $f_3(n) = (-4)^{2n}$ $f_4(n)$ grows asymptotically slower than $f_3(n)$

e. false; for $f_4(n) = (-3)^{(2n)}$ and $f_1(n) = n^2 \lg n$, $f_4 \in \Omega(f_1(n))$, $f_4 \notin O(f_1(n))$, $\therefore f_4 \notin \Theta(f_1(n))$

3. From Ryan Duffy.

3. (e) $f(n) = O([f(n)]^2)$

Suppose $f(n) = \frac{1}{n} \Rightarrow [f(n)]^2 = \left[\frac{1}{n}\right]^2 = \frac{1^2}{n^2} = \frac{1}{n^2} = g(n)$

let $c = 1$

$f(n) \leq c \cdot g(n)$

$\frac{1}{n} \leq \frac{1}{n^2} \Rightarrow n \leq 1$

This implies if $n_0 > 1$ ~~then~~ and $c = 1$ then

$f(n) > cg(n)$ $\therefore$ $f(n) \neq O([f(n)]^2)$

(g) $f(n) = \Theta(f(\frac{n}{2}))$

Suppose $f(n) = e^n \Rightarrow f(\frac{n}{2}) = e^{n/2} = g(n)$

let $c = 1$

$f(n) \leq c \cdot g(n)$

$e^n \leq e^{n/2} \Rightarrow 1 \leq \frac{e^{n/2}}{e^n} = e^{n/2 - n} = e^{n(1/2 - 1)} = e^{-n/2}$

$1 \leq e^{-n/2} \Rightarrow \cancel{\ln(1)}^{0} \leq -\frac{n}{2}$

$0 \leq -\frac{n}{2}$

$0 \leq -n$

$0 \geq n$

If $n_0 > 0$ and $c = 1$ then $f(n) > cg(n)$ $\therefore f(n) \neq O(f(\frac{n}{2}))$ $\Rightarrow f(n) \neq \Theta(f(\frac{n}{2}))$

4. From Jacob Montpetit.

4. Analysis: (10 points) Your client is developing two new algorithms. $f1(n)$ and $f2(n)$ *are* the worst-case running time for these two algorithms: $f1(n) = nlgn$, *and* $f2(n) = 128n$. As a consultant, which algorithm will you recommend to your client? Justify your answer. (Hint: Please consider the asymptotical growth of the functions and also consider the reality.)

$nlg(n) \le 128n$ when n is less or equal to 2^{128} this inequality holds true. When n is greater than 2^{128} 128n gives better performance. I would recommend using the nlg(n) unless n is 2^{128} typically. 2^{128} is approximately 3.4028237e+38, which is a very large number.

Work: $nlg(n) \le 128n$

$$\cancel{n}lg(n) \le 128\cancel{n}$$

$$\cancel{lg}(n) \le 2^{128}$$

$$n \le 2^{128}$$

5. From Etienne Buhrle.

Since this input size is rather unusual, algorithm 1 with running time f_1 is probably still better in most applications.

5 Pseudocode Analysis

For the running time, we get

$$
\begin{aligned}
T_{Mystery}(n) &= c_1 + (n^2+1)c_2 + c_3 \sum_{i=1}^{n^2} (i+1) + c_4 \sum_{i=1}^{n^2} i + c_5 \\
&= c_1 + c_5 + (n^2+1)c_2 + (c_3 + c_4) \sum_{i=1}^{n^2} i + c_3 \sum_{i=1}^{n^2} 1 \\
&= c_1 + c_5 + (n^2+1)c_2 + (c_3 + c_4)\frac{1}{2} n^2 (n^2+1) + c_3 n^2 \\
&= c_1 + c_5 + c_2 n^2 + c_2 + (c_3 + c_4)\frac{1}{2}(n^4 + n^2) + c_3 n^2 \\
&= \Theta(n^4)
\end{aligned}
$$