HW2

Q1.

Credit: Zipei Wei

20 1) Find the order of growth from smallest to largest

From low of Algorithms,

logarithms $\le$ polynomial $\le$ exponential

$lg\left(2^{lg(n^3)}\right) < n^2 lglg(n) < n^2 lg(n) < \left(\frac{1}{3}\right)^{-n} < (-3)^{2n}$

① ② ③ ④ ⑤

a) Compare $lg_2 2^{lg(n^3)} < n^2 lglg(n)$

$log_2 n^3 \le c\, n^2 lglg(n)$ let $c = 1$

$3 \log_2 n_0 = n^2 lglg(n)$

$n_0 \approx 3.76$ (See Pic 1)

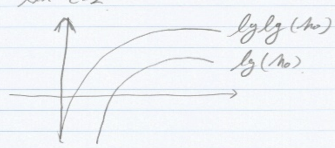
b) compare $n^2 lglg(n) < n^2 lg(n)$

$n^2 lglg(n) \le c \cdot n^2 lg(n)$ let $c=2$

$\cancel{n_0^2} lglg(n_0) \le \cancel{n_0^2} lg(n_0)$

$lglg(n_0) = lg(n_0)$

$lglg(n_0) \le lg(n_0)$



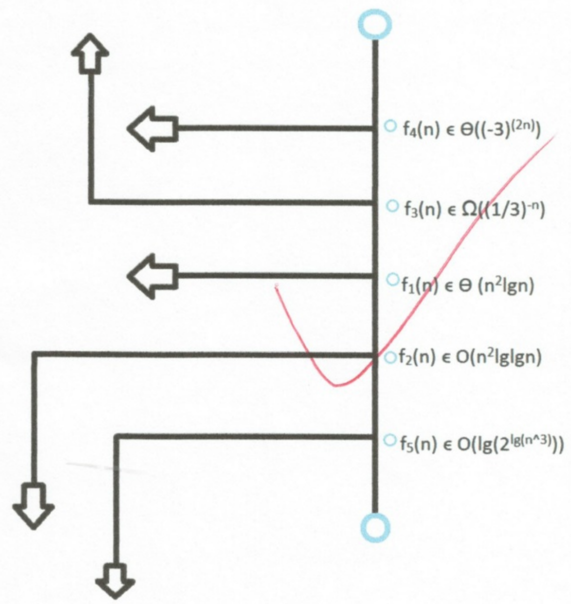
(See Pic 2)

$(n_0 = 16, c = 1)$ $lglg(n_0) = 2 < lg(n_0) = 4$

Q2.

Credit: Molly McGuire

(a) Draw the arrow diagram associated with the 5 statements above



(b) ~ (e) For each statement below, state if it is TRUE (if the statement must always be true, given the assumptions) or FALSE otherwise. In the TRUE case, provide a proof. In the FLASE case, give a counter-example.

b) $f_1(n) \in \Omega(f_2(n))$

Is TRUE because there is no value for n where f_2 is not going to be the lower bound for f_1. f_1 is always the upper bound for f_2, and there is no constant value that can be multiplied to f_1 that would not make it an upper bound.

c) $f_5(n) \in O(f_2(n))$

Is FALSE because given a value of n that is greater than 4.451, f_2 will no longer be the upper bound of f_5.

d) $f_4(n) \in \Omega(f_3(n))$

Is FALSE because when n is less than 0, f_3 is no longer the lower bound of f_4.

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

Is FALSE f_1 is strictly the lower bound of f_4, they are not bounds to one another. There is no value of n or c that would make either of these functions the upper and lower bounds of one another.

Q3.

Credit: YaoLung Su

3. $f(n) = O((f(n))^2)$?

15 False. I will give a counter-example to disprove it.

Let $f(n) = \frac{1}{n}$

$\therefore (f(n))^2 = \left(\frac{1}{n}\right)^2 = \frac{1}{n^2}$

So we try to find c & n_0 to satisfy $0 \le \frac{1}{n} \le c \cdot \frac{1}{n^2}$

we get $n_0 = 1$, $c = 1$

but we cannot find $n > n_0$ to $0 \le \frac{1}{n} \le \frac{1}{n^2}$ (for example: $0 \le \frac{1}{2} \le \frac{1}{4}$ impossible)

$\Rightarrow f(n) = \frac{1}{n} \ne O\left(\frac{1}{n^2}\right)$

Credit: Jason Downing

3. Notation Practice (Textbook Problem 3-4, Part E, Page 62)

15

$$f(n) = O\left((f(n))^2\right)$$

This implies that $f(n) \le c\,(f(n))^2$. However, this can be shown to be false in the case of:

Letting $f(n) = \frac{1}{n^2}$:

$$f(n) = O\left((f(n))^2\right)$$
$$\frac{1}{n^2} = O\left(\left(\frac{1}{n^2}\right)^2\right)$$
$$\frac{1}{n^2} = c\left(\left(\frac{1}{n^2}\right)^2\right)$$
$$\frac{1}{n^2} > c\left(\frac{1}{n^4}\right)$$

This means that when $f(n) = \frac{1}{n^2}$, $f(n) \ne O\left((f(n))^2\right)$.

Q4.

Credit: Therese Kuczynski

4.) I would tell the client that the best algorithm for his application depends on how big his expected n is. As n approaches infinity, $f_2(n)$ (128n) will be less than $f_1(n)$ (nlgn). However, for n where lgn < 128, $f_1(n)$ will be a better algorithm.

Using trial-by-error, the help of a very rough ruby script (as Appendix), I discovered that this n = 340282366920935044470041009808633495552. So if the client is going to be processing amounts of data smaller than the number above (which is almost definitely the case in the real world, but the problem doesn't tell us what exactly he's doing with the data or what dataset he's using), he should use $f_1(n)$ (nlgn), even though it grows asymptotically larger. If, on the other hand, he is processing incredibly large amounts of data as large as or larger then the number above, he should use f2(n), or 128n.

Q5.

Credit: Sony Thach

5) Mystery(n)

25

		Cost for each line	number of execution
1	$c = 0$	C_1	1
2	for $i = 1$ to n^2	C_2	n^2+1
3	for $j = 1$ to i	C_3	$\sum_{i=1}^{n^2} i + 1$
4	$c = c + i - j + n$	C_4	$\sum_{i=1}^{n^2} i$
5	return c	C_5	1

$$T(n) = C_1 + C_2(n^2+1) + C_3\left(\sum_{i=1}^{n^2} i + 1\right) + C_4 \sum_{i=1}^{n^2} i + C_5$$

$$= C_1 + C_2(n^2+1) + C_3\left(\left(\tfrac{1}{2}n^2(n^2+1)\right) + 1\right) + C_4\left(\tfrac{1}{2}n^2(n^2+1)\right) + C_5$$

$$= \left(\frac{C_3+C_4}{2}\right) n^4 + \left(\frac{C_3+C_4}{2}\right) n^2 + C_2 n^2 + (C_1 + C_2 + C_3 + C_5)$$

$$= an^4 + bn^2 + c$$

$$= \Theta(n^4)$$