# Algorithms -- COMP.4040 Honor Statement
(Courtesy of Prof. Tom Costello and Karen Daniels with modifications)

**Must be attached to each submission**

Academic achievement is ordinarily evaluated on the basis of work that a student produces independently. Infringement of this Code of Honor entails penalties ranging from reprimand to suspension, dismissal or expulsion from the University.

Your name on any exercise is regarded as assurance and certification that what you are submitting for that exercise is the result of your own thoughts and study. Where collaboration is authorized, you should state very clearly which parts of any assignment were performed with collaboration and name your collaborators.

In writing examinations and quizzes, you are expected and required to respond entirely on the basis of your own memory and capacity, without any assistance whatsoever except such as what is specifically authorized by the instructor.

I certify that the work submitted with this assignment is mine and was generated in a manner consistent with this document, the course academic policy on the course website on Blackboard, and the UMass Lowell academic code.

Date: 5/29/2019

Name (please print): PHONG VO

Signature: Phong

**Due Date**: May 30, 2019 (Th), BEFORE the lecture starts

This assignment covers textbook Chapter 3 & Chapter1~2.

1. **O, Ω, Θ Notation Practice**: (15 points)
   Provide either a proof (using definition) to support the claim or a counter-example to disprove it.

   (1) $2^{n+1} = O(2^n)$
   (2) $f(n) = \Theta(f(\frac{n}{2}))$
   (3) $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$

2. **Function Order of Growth**: (20 points)
   List the 4 functions below in nondecreasing asymptotic order of growth.

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

   (1) smallest (2) (3) (4) largest

   Justify your answer mathematically by showing values of $c$ and $n_0$ for each pair of functions that are adjacent in your ordering.

3. **O, Ω, Θ Notation Practice**: (30 points, 6 points for each)
   Given (for large n):

   (1) $f_1(n) \in \Omega((\lg n)^2)$ (2) $f_2(n) \in O(n^2 - n)$
   (3) $f_3(n) \in \Omega(\frac{1}{n^2})$ (4) $f_4(n) \in \Theta(\lg(2^{\lg(n^2)}))$

   (a) Draw the arrow diagram associated with the 4 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 FALSE case, give a counter-example.

   (b) $f_4(n) \in O(f_1(n))$
   (c) $f_2(n) \in \Omega(f_3(n))$
   (d) $f_1(n) \in O(f_2(n))$
   (e) $f_4(n) \in \Theta(\lg^3 n)$

4. **Analysis**: (10 points)
   Your client is developing two new algorithms. $f_1(n)$ and $f_2(n)$ *are* the worst-case running time for these two algorithms: $f_1(n) = n\lg n$, *and* $f_2(n) = 512n$. 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.)

5. **Pseudocode Analysis** (25 points)
For the pseudocode below for procedure Mystery($n$), derive tight upper and lower bounds on its asymptotic *worst-case* running time $f(n)$. That is, for the set of inputs including those that force Mystery to work its hardest, find $g(n)$ such that $f(n) \in \Theta(g(n))$. Assume that the input $n$ is a positive integer. Justify your answer.

```
Mystery (n)
if n is an even number
    for i = 1 to n
        for j = n/2 to n
            print "even"
else
    for k = 1 to n/4
        for m = 1 to n
            print "odd"
```

1 (1) $2^{n+1} = O(2^n)$ $\quad 2^{n+1} = 2\cdot 2^n \quad c=2, n_0=1$

- The constant factor doesn't matter and $2^{n+1}$ grows faster than $2^n$ because $2^{n+1} = 2\times 2^n$ is always greater than $2^n$ $\forall n > 0$

Hence, this is a wrong proof. $-5$

(2) $f(n) = \theta(f(\frac{n}{2}))$

The function $f(n)$ grows faster than $f(\frac{n}{2})$ so $f(n)$ is always the upper bound of $f(\frac{n}{2})$. So, $f(n)$ and $f(\frac{n}{2})$ are not the bounds to one another. Hence, there are no $c_{1,2}$ and $n_0$ existed to have $f(\frac{n}{2})$ be the bounds of $f(n)$, such as

$$0 \leq c_1 \cdot f(\tfrac{n}{2}) \leq f_n \leq c_2 \cdot f(\tfrac{n}{2}) \quad , \text{for all } n \geq n_0$$

(3) $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$

| LHS: $f(n) = O(g(n))$ | RHS: $g(n) = \Omega(f(n))$ |
| --- | --- |
| $\Leftrightarrow 0 \leq f(n) \leq c_1 \cdot g(n)$ | $\Leftrightarrow 0 \leq c_2 \cdot f(n) \leq g(n)$ |
| (whereas constants $c, n_0 > 0$) | $\Leftrightarrow 0 \leq f(n) \leq \frac{1}{c_2} g(n)$ |

Let $c_1$ (of LHS) $= \frac{1}{c_2}$ (of RHS),

$\Rightarrow f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$

$\Rightarrow$ proved!

2.

(1) $n^{-2}$ Smallest  (2) $\lg(2^{\lg(n^2)})$  (3) $(\lg n)^2$  (4) $2n^2$ largest

prove: $n^{-2} < \lg(2^{\lg(n^2)})$

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

$n^{-2} = O(2\lg n) \Rightarrow 0 \leq \frac{1}{n^2} \leq c2\lg n$

$\Leftrightarrow 0 \leq 1 \leq 2cn^2 \lg n$

Let $c = \frac{1}{2} \Rightarrow 0 \leq 1 \leq n^2 \lg n$

pick $n_0 = 2$ $\Rightarrow$ proved.

prove: $\underbrace{\lg(2^{\lg(n^2)})}_{=(\lg n^2) = 2\lg n} < (\lg n)^2 \Leftrightarrow 2\lg n < (\lg n)^2$

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

$\Leftrightarrow 0 \leq 2\lg n \leq c.(\lg n)^2$

Let $c = 1 \Rightarrow 0 \leq 2\lg n \leq (\lg n)^2$

$n_0 = 8$

prove: $(\lg n)^2 < 2n^2$

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

$\Rightarrow 0 \leq (\lg n)^2 \leq c \cdot 2n^2$

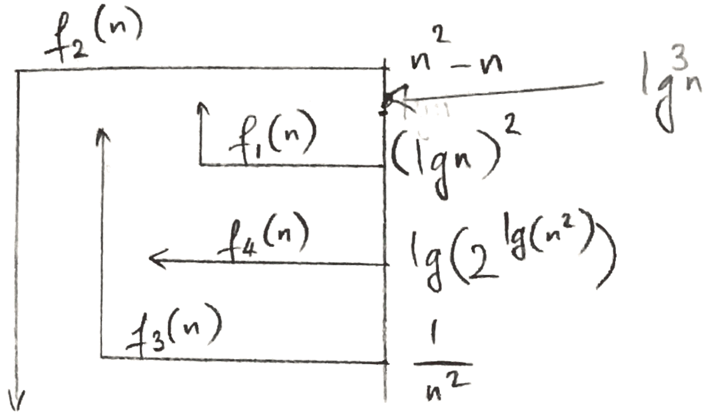
Let $c = 1 \Rightarrow 0 \leq (\lg n)^2 \leq 2n^2$

$n_0 = 2$

HW2/ 3.

(a) Arrow diagram

(3)




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

$\Leftrightarrow \underbrace{\lg(2^{\lg(n^2)})} \in (\lg n)^2$

$= \lg(n^2) = 2\lg n \in (\lg n)^2$

$\boxed{\text{TRUE}}$ because $f_1(n)$ always grows faster than $f_4(n)$ with all values of $n$.

(c) $f_2(n) \in \Omega(f_3(n)) \Leftrightarrow n^2-n \geq c \cdot \frac{1}{n^2}$

Let $c=1 \Rightarrow n^2-n \geq \frac{1}{n^2}$

If $n_0 = 1 \Rightarrow f_2 = 0 \geq f_3 = 1$ is $\boxed{\text{FALSE}}$

Besides, there be some values of $n$ existed that make $f_3(n)$ can be greater than $f_2(n)$.

(d) $f_1(n) \in O(f_2(n)) \Leftrightarrow (\lg n)^2 \leqslant c \cdot (n^2 - n)$

Let $c = 1 \Rightarrow (\lg n)^2 \leqslant n^2 - n$

If $n_0 = \frac{1}{4} \Rightarrow \quad 4 \leqslant \frac{1}{16} - \frac{1}{4} = \frac{-3}{16} \Rightarrow \boxed{\text{FALSE}}$

(e) $f_4(n) \in \Theta(\lg^3 n) \Leftrightarrow \lg(2^{\lg n^2}) \in \lg^3 n$

$\Leftrightarrow \lg n^2 \in \lg^3 n \Leftrightarrow 2\lg n \in \lg^3 n$

$\Leftrightarrow c_1 \lg^3 n \leqslant 2\lg n \leqslant c_2 \lg^3 n$

- Case 1 : $c_1 \lg^3 n \leqslant 2\lg n$

  Let $c_1 = 1 \Rightarrow \lg^3 n \leqslant 2\lg n$

  $n_0 = 4 \Rightarrow \quad 8 \leqslant 4 \Rightarrow \boxed{\text{FALSE}}$

- case 2: $2\lg n \leqslant c_2 \cdot \lg^3 n$

  Let $c_2 = 1 \Rightarrow 2\lg n \leqslant \lg^3 n$

  $n_0 = 2 \Rightarrow \quad 2 \leqslant 1 \Rightarrow \boxed{\text{FALSE}}$

Combine the two conclusions of case 1 and case 2 $\Rightarrow \boxed{\text{FALSE}}$

Besides, the function $\lg^3 n$ is an upper bound of $f_4(n)$ but they are not bounds to one another.

HW2 (4)

(4) Consider: $n \lg n \leq 512n$

$\Rightarrow \lg n \leq 512 \Rightarrow n \leq 2^{512}$

Hence, $n \lg n$ is less than $512n$ whereas $n \leq 2^{512}$.

So, I'd recommend the function $n \lg n$ algorithm if working with $n \geq 2^{512}$

(5)

| | | |
|---|---|---|
| $c_1$ | $1$ | 1. if $n$ is an even number |
| $c_2$ | $n+1$ | 2. for $i = 1$ to $n$ |
| $c_3$ | $(\frac{n}{2}+1)n$ | 3. for $j = n/2$ to $n$ |
| $c_4$ | $(\frac{n}{2})n$ | 4. print "even" |
| $c_5$ | $0$ | 5. else |
| $c_6$ | $\frac{n}{4}+1$ | 6. for $k = 1$ to $n/4$ |
| $c_7$ | $(n+1)\frac{n}{4}$ | 7. for $m = 1$ to $n$ |
| $c_8$ | $n \cdot (\frac{n}{4})$ | 8. print "odd" |

$$T(n) = c_1 \cdot 1 + c_2(n+1) + c_3\left(\frac{n}{2}+1\right)n + c_4\left(\frac{n}{2}\right)n + c_6\left(\frac{n}{4}+1\right) + c_7(n+1)\frac{n}{4} + c_8 \times n \times \frac{n}{4}$$

$$= c_1 + c_2 n + c_2 + c_3\frac{n^2}{2} + c_3 n + c_4\frac{n^2}{2} + c_6\frac{n}{4} + c_6 + c_7\frac{n^2}{4} + c_7\frac{n}{4} + c_8\frac{n^2}{4}$$

$$= n^2\left(\frac{c_3}{2} + \frac{c_4}{2} + \frac{c_7}{4} + \frac{c_8}{4}\right) + n\left(c_2 + c_3 + \frac{c_6}{4} + \frac{c_7}{4}\right)$$

$$+ (c_1 + c_2 + c_6)$$

$\Rightarrow$ Quadratic functional algorithm $\Rightarrow \theta(n^2)$

95/100