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: 02/14/2019

Name (please print): DANGNHI NGO

Signature:

Due Date: Feb. 15, 2019 (F), BEFORE the lecture starts

This assignment covers textbook Chapter 3 & Chapter1~2.

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

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

   (1) smallest $\qquad$ (2) $\qquad$ (3) $\qquad$ (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.

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

   (1) $f_1(n) \in \Omega((\lg n)^3)$ $\qquad$ (2) $f_2(n) \in O(n^3 - \frac{1}{n})$

   (3) $f_3(n) \in \Omega(\frac{1}{n^3})$ $\qquad$ (4) $f_4(n) \in \Theta(\lg(2^{\lg(n^3)}))$

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

3. **O, Ω, Θ Notation Practice**: (15 points)
   Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation to prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

   Hint: $\max(f(n), g(n)) = \begin{cases} f(n), if\ f(n) \geq g(n) \\ g(n), if\ f(n) < g(n) \end{cases}$

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* $f_2(n) = 256n$. 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

$\frac{n^2}{3^{i-1}} = 1$ $\qquad n^2 = 3^{i-1}$

$i = \lg_3 n^2 + 1$

$(\lg_3 n^2 + 1)\, cn^2$

$cn^2 \lg_3 n^2 + cn^2 \qquad O(n^2)$

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 downto n/2
            print "1"
else
    for k = 1 to n/4
        for m = 1 to n
            print "2"
```

20

1/ Function Order of Growth

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

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

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

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

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

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

$n_0 = 2$

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

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

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

Let $c = 1$, $0 \leq \lg(n^3) \leq (\lg n)^3$

$n_0 = 4$

c/ Compare $(\lg n)^3 < n^3$

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

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

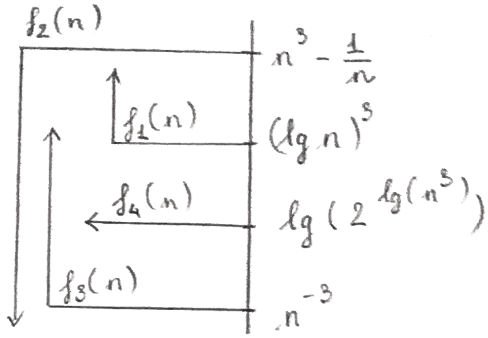
Let $c = 1$, $0 \leq (\lg n)^3 \leq n^3$

$n_0 = 1$

30

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.

15 3/ O, Ω, Θ Notation Practice

Prove: $\max(f(n), g(n)) = \Theta(f(n) + g(n))$

Assume that $f(n)$ and $g(n)$ be asymptotically nonnegative functions

There exists n_1, n_2 that $f(n) \geq 0$ with $n > n_1$

and $g(n) \geq 0$ when $n > n_2$

$$\left.\begin{array}{l} \max(f(n), g(n)) \geq f(n) \\ \max(f(n), g(n)) \geq g(n) \end{array}\right\} \text{ for } n > n_0 \text{ and } n_0 = \max(n_1, n_2)$$

$\Rightarrow 2\max(f(n), g(n)) \geq f(n) + g(n)$

$\Rightarrow O(\max(f(n), g(n)) = f(n) + g(n)$ (1)

Besides, $$\left.\begin{array}{l} f(n) \leq f(n) + g(n) \\ g(n) \leq f(n) + g(n) \end{array}\right\}$$

$\Rightarrow \max(f(n), g(n)) \leq f(n) + g(n)$

$\Rightarrow \Omega(\max(f(n), g(n)) = f(n) + g(n)$ (2)

From (1) and (2), $\Theta(\max(f(n), g(n)) = f(n) + g(n)$ (Theorem of Θ-notation)

$\Rightarrow \max(f(n), g(n)) = \Theta(f(n) + g(n))$ (Symmetric property of Θ)

100/100

4/ Analysis

10

$f_1(n) = n\lg(n)$

$f_2(n) = 256n$

$n\lg(n) \leqslant 256.n$ (because logarithm < linear)

$\Leftrightarrow \lg(n) \leqslant 256$

$\Leftrightarrow n \leqslant 2^{256}$

Conclusion: As a consultant, I would recommend my client to use $f_1(n) = n\lg(n)$ unless $n > 2^{256}$. Because when $n > 2^{256}$, $f_2(n) = 256.n$ gives better performance.

$2^{256} \approx 1.15792\,e^{77}$, a very large number.

5/ Pseudocode Analysis

25

$T_{Mystery}(n) = c + n + \frac{n}{2} + \frac{cn^2}{2}$ (if n is an even number)

$T_{Mystery}(n) = c + \frac{n}{4} + n + \frac{cn^2}{4}$ (if n is an odd number)

$O\left(c + n + \frac{n}{2} + \frac{cn^2}{2}\right) = O(n^2)$

$\Omega\left(c + \frac{n}{4} + n + \frac{cn^2}{4}\right) = \Omega(n^2)$

$\Rightarrow$ For the sets of inputs including those that force Mystery to work its hardest, total run time $= \Theta(n^2)$ (n is a positive integer)

Mystery (n)		Running time
if n is an even number	C1	1
for i = 1 to n	C2	$n+1$
for j = n down to $\frac{n}{2}$	C3	$n\left(\frac{n}{2}+1\right)$
print "1"	C4	$n.\frac{n}{2}$
else	C5	
for k = 1 to $\frac{n}{4}$	C6	$\frac{n}{4}+1$
for m = 1 to n	C7	$\frac{n}{4}(n+1)$
print "2"	C8	$\frac{n}{4}.n$