University of Waterloo
CS240 - Spring 2014
Assignment 1

Due Date: Wednesday May 21 at 09:15am

Please read `http://www.student.cs.uwaterloo.ca/~cs240/s14/guidelines.pdf` for guidelines on submission. Problems 1 – 6 are written problems; submit your solutions electronically as a PDF with file name `a01wp.pdf` using MarkUs. We will also accept individual question files named `a01q1w.pdf`, `a01q2w.pdf`, ..., `a01q6w.pdf` if you wish to submit questions as you complete them.

There are 78 marks available; the assignment will be marked out of 74.

Problem 1 [4+4+4+4+4=20 marks]

Provide a complete proof of the following statements from first principles (i.e., using the original definitions of order notation). All logarithms are natural logarithms: log = ln.

1. $n^3 - 21n^2 + 100 \in O(n^3)$

$$
\begin{aligned}
&n^3 - 21n^2 + 100 \in O(n^3) \leq cn^3 \\
&n^3 + 100 \geq n^3 - 21n^2 + 100 \\
&n^3 + 100 \leq cn^3, \text{ let c = 2} \\
&100 \leq n^3 \implies \sqrt[3]{100} \leq n \\
&n_0 = \sqrt[3]{100}, c = 2 \\
&\therefore n^3 - 21n^2 + 100 \in O(n^3)
\end{aligned}
$$

2. $(n+10)^3 \in \Theta(n^3)$

$$
\begin{aligned}
&\text{Prove: } (n+10)^3 \in O(n^3)\\
&(n+10)^3 \leq cn^3\\
&n^3 + 30n^2 + 300n + 1000 \leq cn^3\\
&\text{let c} = 4\\
&n^3 \leq n^3\\
&30n^2 \leq n^3 \implies n \geq 30\\
&300n \leq n^3 \implies n \geq \sqrt(300)\\
&1000 \leq n^3 \implies n \geq \sqrt[3](1000)\\
&n_0 = 30, c = 4\\
&\therefore (n+10)^3 \in O(n^3)
\end{aligned}
$$

$$
\begin{aligned}
&\text{Prove: } (n+10)^3 \in \Omega(n^3)\\
&(n+10)^3 \geq cn^3\\
&n+10 \geq c_1 n \text{ where } c_1 = c^{1/3}\\
&n_0 = 1, c_1 = 1\\
&\therefore (n+10)^3 \in \Omega(n^3)
\end{aligned}
$$

$$\therefore (n+10)^3 \in \theta(n^3)$$

3. $1000n \in o(n \log n)$

$$
\begin{aligned}
&1000n \leq cn\log(n)\\
&1000 \leq c\log(n)\\
&\frac{1000}{c} \leq log(n)\\
&n \geq e^{\frac{1000}{c}}\\
&n_0 = e^{\frac{1000}{c}} \quad \forall c > 0\\
&\therefore (n+10)^3 \in o(n\log n)
\end{aligned}
$$

4. $n! \in o(n^n)$ where $n! = \prod_{i=1}^{n} i$

$$n! = n(n-1)...(1)$$
$$n(n-1)...(1) \leq n^{n-1}$$
$$n^{n-1} \leq cn^n$$
$$1 \leq cn$$
$$n_0 \geq \frac{1}{c} \quad \forall c > 0$$
$$\therefore n! \in o(n^n)$$

5. Bonus question : Let $H(n) = \sum_{i=1}^{n} 1/i$. Prove that $H(n) \in \omega(1)$.

$$Prove : \sum_{i=1}^{n} \frac{1}{i} \geq c(1)$$
$$\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + (\frac{1}{3} + \frac{1}{4}) + (\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}) \geq 1 + \sum_{i=1}^{log_2(n)} \frac{1}{2}$$
$$\implies 1 + \frac{1}{2} log_2(n) \geq c$$
$$\implies \frac{1}{2} log_2(n) \geq c$$
$$\implies n \geq 2^{2c}$$
$$\implies \exists\, n_0 > 0 \cdot \quad \forall c > 0 \; n_0 \geq 2^{2c} \quad \forall n > n_0$$
$$\therefore H(n) \in \omega(1)$$

Problem 2 [4+4+4=12 marks]

For each pair of the following functions, fill in the correct asymptotic notation among Θ, o, and ω in the statement $f(n) \in \sqcup(g(n))$. Provide a brief justification of your answers. In your justification you may use any relationship or technique that is described in class.

(a) $f(n) = n^3(5 + 2\cos 2n)$ versus $g(n) = 3n^2 + 4n^3 + 5n$

$2cos(2\pi)$ is bounded by -2 and $2 \implies n^3(5 + 2\cos 2n)$ is bounded by $3n^3$ and $7n^3$
let $c_1 = 3$ and $c_2 = 7$
$\therefore f(n) \in \theta(n^3)$

$\theta(3n^2 + 4n^3 + 5n) = \theta(4n^3)$ by Max Rule
$\theta(4n^3) = \theta(n^3) \implies g(n) \in \theta n^3$
$\therefore f(n) \in \theta g(n)$

(b) $f(n) = n(\log n)^3$ versus $g(n) = n^2$. *Hint:* Use L'Hopital's rule.

$$
\begin{aligned}
&lim_{n\to\infty}\frac{n(log_n)^3}{n^2} \to \infty \text{ as n } \to \infty \\
&= lim_{n\to\infty}\frac{3n(log_n)^2\frac{1}{n}}{2n} \\
&= lim_{n\to\infty}\frac{2(logn)\frac{1}{n}}{2n} \\
&= lim_{n\to\infty}\frac{2}{4n^2} \to 0 \text{ as n } \to 0 \\
&\therefore f(n) \in o(g(n))
\end{aligned}
$$

(c) $f(n) = n^{0.01}$ versus $g(n) = (\log n)^2$

$$
\begin{aligned}
&lim_{n\to\infty}\frac{f(n)}{g(n)} \to \infty \text{ as n } \to \infty \\
&= lim_{n\to\infty}\frac{n^{0.01}}{(log(n)^2)} \\
&= lim_{n\to\infty}\frac{0.01n^{-0.99}}{2(log(n)\frac{1}{n})} \\
&= lim_{n\to\infty}\frac{0.01n^{0.01}}{2log(n)} \\
&= lim_{n\to\infty}\frac{0.0001n^{-0.99}}{\frac{2}{n}} \\
&= lim_{n\to\infty}\frac{0.0001n^{0.01}}{2} \to \infty \text{ as n } \to \infty
\end{aligned}
$$

Problem 3 [4+4+4+4+4=20 marks]

Prove or disprove each of the following statements. To prove a statement, you should provide a formal proof that is based on the definitions of the order notations. To disprove a statement, you can either provide a counter example and explain it or provide a formal proof. All functions are positive functions.

(a) $f(n) \notin \omega(g(n)) \Rightarrow f(n) \in O(g(n))$

$$
\begin{aligned}
&!(\ \forall c > 0 \ \exists\ n_0 > 0 \cdot 0 \leq cg(n) \leq f(n) \quad \forall n > n_0) \text{ and } g(n) > 0 \text{ and } f(n) > 0 \\
&\implies \exists\ c > 0 \ \cdot \quad \forall n_0 > 0 \ \exists\ n > n_0 \cdot \ (c(g(n)) < 0 \text{ or } c(g(n)) > f(n)) \text{ and } g(n) > 0 \text{ and } f(n) > 0 \\
&\implies \exists\ c > 0 \ \cdot \quad \forall n_0 > 0 \ \exists\ n > n_0 \cdot \ (c(g(n)) > f(n) > 0) \\
&\implies \exists\ c, n_0 > 0 \ \cdot \ (c(g(n)) > f(n) > 0 \quad \forall n > n_0) \\
&\implies f(n) \in O(g(n))
\end{aligned}
$$

(b) $f(n) \in O(g(n)) \Rightarrow \exists\ c > 0\ \forall n \in \mathbb{N}, f(n) < cg(n)$

$$
\begin{aligned}
&f(n) \leq f(n_0)\ \forall n \leq n_0 \\
&f(n) \leq cg(n)\ \forall n \geq n_0 \\
&c(g(n)) + f(n_0) \geq f(n)\ \forall n < n_0 \\
&\text{since } f(n) < f(n_0) < \frac{g(n)f(n_0)}{g(0)}\ \forall n < n_0 \\
&c(g(n)) + \frac{g(n)f(n_0)}{g(0)} \geq f(n)\ \forall n < n_0 \\
&\implies g(n)(c + \frac{f(n_0)}{g(0)}) \geq f(n)\ \forall n < n_0 \\
&\text{let } c_1 = c + \frac{f(n_0)}{g(0)} \\
&f(n) \leq c_1(g(n))\ \forall n < n_0 \text{ and } f(n) \leq cg(n)\ \forall n \geq n_0 \implies f(n) \leq c_1 g(n)\ \forall n > 0
\end{aligned}
$$

$\therefore f(n) \in O(g(n)) \Rightarrow \exists\ c > 0\ \forall n \in \mathbb{N}, f(n) < cg(n)$

(c) $f(n) \in \Theta(g(n)) \Rightarrow 2^{f(n)} \in \Theta(2^{g(n)})$

$$Prove : 2^{f(n)} \in O(2^{g(n)})$$

$$\exists\ c_1, n_0 > 0 \cdot 0 \leq f(n) \leq c_1 g(n) \quad \forall n \geq n_0$$

$$\implies 0 \leq 2^{f(n)} \leq 2^{c_1 g(n)} \quad \forall n \geq n_0$$

$$\implies 0 \leq 2^{f(n)} \leq 2^{(c_1-1)g(n_0)} 2^{g(n)} \quad \forall n \geq n_0$$

$$\text{let } c_2 = 2^{(c_1-1)g(n_0)}$$

$$\implies 0 \leq 2^{f(n)} \leq c_2 2^{g(n)} \quad \forall n \geq n_0$$

$$\therefore 2^{f(n)} \in O(2^{g(n)})$$

$$Prove : 2^{f(n)} \in \Omega(2^{g(n)})$$

$$\exists\ c_1, n_0 > 0 \cdot 0 \leq c_1 g(n) \leq f(n) \quad \forall n \geq n_0$$

$$\implies 0 \leq 2^{c_1 g(n)} \leq 2^{f(n)} \quad \forall n \geq n_0$$

$$\implies 0 \leq 2^{(c_1-1)g(n_0)} 2^{g(n)} \leq 2^{f(n)} \quad \forall n \geq n_0$$

$$\text{let } c_2 = 2^{(c_1-1)g(n_0)}$$

$$\implies 0 \leq c_2 2^{g(n)} \leq 2^{f(n)} \quad \forall n \geq n_0$$

$$\therefore 2^{f(n)} \in \theta(2^{g(n)})$$

(d) $f(n) \in \Theta(g(n))$ and $h(n) \in \Theta(g(n)) \Rightarrow \frac{f(n)}{h(n)} \in \Theta(1)$

$$Prove: \frac{f(n)}{h(n)} \in O(1)$$
$$\exists\ c_1, n_0 > 0 \cdot 0 \leq f(n) \leq c_1 g(n) \quad \forall n \geq n_0$$
$$\exists\ c_1, n_1 > 0 \cdot 0 \leq h(n) \leq c_2 g(n) \quad \forall n \geq n_1$$

$$\frac{f(n)}{h(n)} \leq \frac{c_1 g(n)}{c_2 g(n)} \leq c_3(1)$$
$$\therefore \frac{f(n)}{h(n)} \in O(1)$$

$$Prove: \frac{f(n)}{h(n)} \in \Omega(1)$$
$$\exists\ c_1, n_0 > 0 \cdot 0 \leq f(n) \geq c_1 g(n) \quad \forall n \geq n_0$$
$$\exists\ c_1, n_1 > 0 \cdot 0 \leq h(n) \geq c_2 g(n) \quad \forall n \geq n_1$$

$$\frac{f(n)}{h(n)} \geq \frac{c_1 g(n)}{c_2 g(n)} \geq c_3(1)$$
$$\therefore \frac{f(n)}{h(n)} \in \Omega(1)$$

$$\therefore \frac{f(n)}{h(n)} \in \theta(1)$$

(e) $\min(f(n), g(n)) \in \Theta\left(\frac{f(n)g(n)}{f(n)+g(n)}\right)$

Prove: $min(f(n), g(n)) \in O(\dfrac{f(n)g(n)}{f(n)+g(n)})$

$$min(f(x), g(x)) = \frac{f(x)+g(x)-|f(x)-g(x)|}{2}$$
$$\implies \exists\ c, x_0 > 0 \cdot \ \frac{f(x)+g(x)-|f(x)-g(x)|}{2} \leq c\frac{f(x)g(x)}{f(x)+g(x)}\ \forall x \geq x_0$$

Case: $f(x) > g(x)$

$$\implies g(x) \leq \frac{cf(x)g(x)}{f(x)+g(x)} \; \forall x \geq x_0$$

$$\implies 1 \leq \frac{cf(x)}{f(x)+g(x)} \; \forall x \geq x_0$$

$$\implies f(x)+g(x) \leq cf(x) \; \forall x \geq x_0$$

$$\implies 2f(x) \leq cf(x) \; \forall x \geq x_0$$

$$\implies 2 \leq c \; \forall x \geq x_0$$

where $c = 3$ and $x_0 = 1$

Case: $f(x) < g(x)$

$$\implies f(x) \leq \frac{cf(x)g(x)}{f(x)+g(x)} \; \forall x \geq x_0$$

$$\implies 1 \leq \frac{cg(x)}{f(x)+g(x)} \; \forall x \geq x_0$$

$$\implies f(x)+g(x) \leq cg(x) \; \forall x \geq x_0$$

$$\implies 2g(x) \leq cg(x) \; \forall x \geq x_0$$

$$\implies 2 \leq c \; \forall x \geq x_0$$

where $c = 3$ and $x_0 = 1$

$$\therefore min(f(n), g(n)) \in O(\frac{f(n)g(n)}{f(n)+g(n)})$$

Prove: $min(f(n), g(n)) \in \Omega(\frac{f(n)g(n)}{f(n)+g(n)})$

$$min(f(x), g(x)) = \frac{f(x)+g(x)-|f(x)-g(x)|}{2}$$

$$\implies \exists \; c, x_0 > 0 \cdot \; \frac{f(x)+g(x)-|f(x)-g(x)|}{2} \geq c\frac{f(x)g(x)}{f(x)+g(x)} \; \forall x \geq x_0$$

$$\text{Case: } f(x) > g(x)$$
$$\implies g(x) \geq \frac{cf(x)g(x)}{f(x)+g(x)} \; \forall x \geq x_0$$
$$\implies 1 \geq \frac{cf(x)}{f(x)+g(x)} \; \forall x \geq x_0$$
$$\implies f(x)+g(x) \leq cf(x) \; \forall x \geq x_0$$
$$\implies 2f(x) \geq cf(x) \; \forall x \geq x_0$$
$$\implies 2 \geq c \; \forall x \geq x_0$$
$$\text{where } c = 1 \text{ and } x_0 = 1$$

$$\text{Case: } f(x) < g(x)$$
$$\implies f(x) \geq \frac{cf(x)g(x)}{f(x)+g(x)} \; \forall x \geq x_0$$
$$\implies 1 \geq \frac{cg(x)}{f(x)+g(x)} \; \forall x \geq x_0$$
$$\implies f(x)+g(x) \geq cg(x) \; \forall x \geq x_0$$
$$\implies 2g(x) \geq cg(x) \; \forall x \geq x_0$$
$$\implies 2 \geq c \; \forall x \geq x_0$$
$$\text{where } c = 3 \text{ and } x_0 = 1$$
$$\therefore min(f(n), g(n)) \in \Omega(\frac{f(n)g(n)}{f(n)+g(n)})$$

$\therefore min(f(n), g(n)) \in \Theta(\frac{f(n)g(n)}{f(n)+g(n)})$

Problem 4 [4+4+2=10 marks]

Let θ be either 2 or 3. Consider the sum

$$f(n) := \sum_{i=0}^{\log_2 n} 4^i \left\lfloor \frac{n}{2^i} \right\rfloor^\theta .$$

(a) Assume that n is a power of two.

Give an exact closed form for $f(n)$ in terms of n and θ.
Hint: Re-write the formula as a geometric series, and treat $\theta = 2$ and $\theta = 3$ as separate

cases.

$\theta = 2$ case:

$$\sum_{i=0}^{log_2 n} 4^i \lfloor \frac{n}{2^i} \rfloor^{\theta}$$

$$= \sum_{i=0}^{log_2 n} 2^{2i} \frac{n^{\theta}}{2^{(\theta)i}} \text{ The floor can be dropped because n is a power of 2}$$

$$= \sum_{i=0}^{log_2 n} n^{\theta} 2^{2i-(\theta)i}$$

Since when $\theta = 2, r = 1$, use the geometric series formula:

$$\sum_{i=0}^{n-1} ar^i = na$$

$$\implies \sum_{i=0}^{log_2 n} n^{\theta} 2^{2i-(\theta)i} = (log_2 n + 1)n^{\theta}$$

$\theta = 3$ case:

$$\sum_{i=0}^{log_2 n} 2^{2i-\theta(i)} n^{\theta}$$

Since when $\theta = 3, 0 < r < 1$, use the geometric series formula:

$$\sum_{i=0}^{n-1} ar^i = a\frac{1-r^n}{1-r}$$

$$\implies \sum_{i=0}^{log_2 n} 2^{2i-\theta(i)} n^{\theta} = n^{\theta}(1 - \frac{1-(2^{2-\theta})^{log_2 n+1}}{1-2^{2-\theta}})$$

(b) Now consider the function $g(n) := f(2^{\lceil \log_2 n \rceil})$, defined for all positive n. Give simple bounds for $g(n)$ using Θ-notation. You should have two Θ bounds: one for $\theta = 2$ and one for $\theta = 3$.

(c) Deduce simple bounds on $f(n)$ using Θ-notation.

when $\theta = 2$

$$f(n) \in \theta(n^{\theta}(logn))$$

when $\theta = 3$

$$f(n) \in \theta(n^{\theta}(1 - \frac{1-(2^{2-\theta})^{log_2 n+1}}{1-2^{2-\theta}}))$$

Problem 5 [4+4=8 marks]

(a) Prove that the following code fragment will always terminate.
If s is even, then it will take 1 operation before s will become half its value. If s is always even, this will terminate after $log_2 n$ operations, since the loop exists when s is less than 1. Alternatively, if s is odd, then it will take 2 operations before s will become half its value. if the division of s by 2 always resulted in an odd value, then it would take $2log_2$ operations to complete before s will be less than 1. In either case of s being an even or odd number, the program will terminate.

(b) Prove that its running time is $O(\log n)$.

Let c be a constant representing cost of executing up to 2 operations two operations and T(n) be the total cost of arithmetic operations.

$$T(n) = \sum_{i=1}^{n} c$$
$$= clog_2 n$$
$$\therefore T(n) \in \theta(log_2 n)$$

```
s := n  // n is an integer
while (s>1)
   if (s is even)
      s := s/2
   else
      s := s+1
```

Problem 6 [4+4=8 marks]

Consider the following (not necessarily the best) implementation of an algorithm that finds the largest element in the array. Give the best case (4 marks) and worst case (4 marks) running time of the function, using Θ-notation. Justify your answer.

```
function max-element(A[1..n])
    if n = 1 do
        return A[1]
    else if A[1] > max-element(A[2..n]) do
        return A[1]
    else return max-element(A[2..n])
```

Best Case: $n = 1, O(n)$

If the max is the first number, such as in [n..,3,2,1], then the program executes n return operations.

Worst Case: $n = [1, 2, 3, 4], O(n^2)$

The worst performance occurs when the largest element is to the far right. In this instance, the algorithm must run first 4, then 3, 2 and 1 times, each time removing an element from the beginning of the array. Let $T(n)$ model the total cost of the operations. Then the total operations executed in the worst case would be:

$$
\begin{aligned}
T(n) &\in \sum_{i=1}^{n} \sum_{j=i}^{n} \theta(1) \\
&= \sum_{i=1}^{n} (n - i + 1)\theta(1) \\
&= \theta(1) \sum_{i=1}^{n} i \\
&= \theta(1) \frac{(n+1)(n)}{2} \\
&= \theta(1) \frac{n^2 + n}{2} \\
&= \theta(n^2)
\end{aligned}
$$