University of Waterloo

CS240 Winter 2018
Assignment 1

Unless indicated otherwise, logarithms are base 2, i.e. $\log = \log_2$ and the function $\log^2 n$ means $(\log n)^2$.

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

Note: There are a wide variety of solutions for these questions.

a) $8n^4 - 2n^2 + 1 \in \Theta(n^4)$

Let $n_0 = 1$. Let $c_1 = 6$ and let $c_2 = 9$. For $n \geq n_0$ we have

$$\begin{aligned} c_1|n^4| &\leq |6n^4| \\ &\leq |6n^4 + 1| \\ &\leq |6n^4 + (2n^4 - 2n^4) + 1| \\ &\leq |6n^4 + (2n^4 - 2n^2) + 1| \\ &\leq |8n^4 - 2n^2 + 1| \end{aligned}$$

and

$$\begin{aligned} |8n^4 - 2n^2 + 1| &\leq |8n^4 + 1| \\ &\leq |8n^4 + n^4| \\ &\leq |9n^4| \\ &\leq 9|n^4| \\ &\leq c_2|n^4| \end{aligned}$$

Hence $8n^4 - 2n^2 + 1 \in \Theta(n^4)$.

b) $4^{\log n} \in O(n \log^2 n)$

False.
$4^{\log n} = n^{\log 4} = n^2 \notin O(n \log^2 n)$.

c) $n \log n \in \Omega(\sqrt{n})$

Let $n_0 = 2$ and let $c = 1$. For $n \geq n_0$ we have

$$\begin{aligned} c|\sqrt{n}| &= |\sqrt{n}| \\ &\leq |\sqrt{n}\sqrt{n}| \\ &\leq |\sqrt{n}\sqrt{n}\log n| \\ &\leq |n \log n| \end{aligned}$$

Hence $n \log n \in \Omega(\sqrt{n})$.

d) $n \log(4^n) \in \Omega(n^2)$

By the property of logs we have: $n \log(4^n) = n^2 \log 4 = 2n^2$.

Let $n_0 = 1$ and let $c = 1$. For $n \geq n_0$ we have

$$\begin{aligned} c|n^2| &= |n^2| \\ &\leq |n \log(4^n)| \end{aligned}$$

Hence $n \log(4^n) \in \Omega(n^2)$.

e) $n \log(4^n) \in \omega(n^2)$

False.
As above $n \log(4^n) = 2n^2$ and $c|n^2| \not< |2n^2|$ when $c \geq 2$ for any $n > 0$.

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

Note: There are a wide variety of solutions for these questions.

a) $f(n) = \log_a n$ versus $g(n) = \log_b n$ where $a, b > 1$

For logarithms, $\log_a n = \log_b n / \log_b a$ for all $n > 0$. This is based on $\log_b(a) = \frac{\log_c a}{\log_c b}$ from the end of module 1, with suitable substitutions.

Letting c_1 and c_2 equal $1/\log_b a$ which is positive we have $c_1 \log_b n \leq \log_a n \leq c_2 \log_b n$ for all $n > 0$. Hence by the definition of Θ we have $\log_a n \in \Theta(\log_b n)$.

b) $f(n) = 2^{2n}$ versus $g(n) = 2^n$

$f(n) = \omega(g(n))$

Since $2^{2n} = 2^n 2^n$ we have

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{2^n 2^n}{2^n} = \lim_{n\to\infty} 2^n = \infty$$

Hence by the theorem presented in lecture, $f(n) = \omega(g(n))$.

c) $f(n) = n^{\frac{3}{2}}$ versus $g(n) = n \log n$

$f(n) = \omega(g(n))$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^{\frac{3}{2}}}{n \log n} = \lim_{n\to\infty} \frac{n^{\frac{1}{2}}}{\log n} = \frac{\infty}{\infty}$$

Using l'Hopital's rule we have

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)} = \lim_{n\to\infty} \frac{\frac{1}{2}n^{-\frac{1}{2}}}{\frac{1}{\ln 2}\frac{1}{n}} = \lim_{n\to\infty} \frac{\ln 2}{2} n^{\frac{1}{2}} = \infty$$

Hence by the theorem presented in lecture, $f(n) = \omega(g(n))$.

Question 3 [4+4+4=12 marks]

a) If $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$ then $f(n) \in O(h(n))$

Since $f(n) \in O(g(n))$, there exists positive constants c_f and n_f such that $0 \le f(n) \le c_f g(n)$ for all $n \ge n_f$.

Since $g(n) \in O(h(n))$, there exists positive constants c_g and n_g such that $0 \le g(n) \le c_g h(n)$ for all $n \ge n_g$.

Let $n_0 = \max\{n_f, n_g\}$ and $c = c_f c_g$. Since c is the product of two positive numbers, it is also positive. We have $0 \le f(n) \le c_f g(n) \le c_f c_g h(n) = ch(n)$ for all $n \ge n_0$.

Hence $f(n) \in O(g(n))$.

b) If $f(n) \in \Theta(h(n))$ and $g(n) \in \Theta(h(n))$ then $f(n) + g(n)$ is $\Theta(h(n))$.

Since $f(n) \in \Theta(h(n))$, we can find some $c_{f_1}, c_{f_2}, n_f > 0$ such that

$$c_{f_1} h(n) \le f(n) \le c_{f_2} h(n) \tag{1}$$

for all $n > n_f$. Similarly for $g(n)$, we can find some $c_{g_1}, c_{g_2}, n_g > 0$ such that

$$c_{g_1} h(n) \leq g(n) \leq c_{g_2} h(n) \tag{2}$$

Adding (1) and (2) we have:

$$c_{f_1} h(n) + c_{g_1} h(n) \leq f(n) + g(n) \leq c_{f_2} h(n) + c_{g_2} h(n)$$
$$(c_{f_1} + c_{g_1}) h(n) \leq f(n) + g(n) \leq (c_{f_2} + c_{g_2}) h(n)$$

for all $n > \max\{n_f, n_g\}$. Thus $f(n) + g(n) \in \Theta(h(n))$ as desired.

c) If $f(n) \in \Theta(h(n))$ and $g(n) \in \Theta(h(n))$ then $f(n) - g(n)$ is $O(1)$.

Let $f(n) = 2n$ and let $g(n) = n$. Then $f(n) - g(n) = n$ which is unbounded and thus not in $O(1)$. Thus $f(n) - g(n) \notin O(1)$.

Question 4 [4+4+4+4=16 marks]

a)
```
for i = 1 to n
  for j = 1 to log(n)
    for k = 1 to 240
      x = x + 1
```

The innermost for-loop iterates a constant number of times. The middle for-loop iterates $\log(n)$ times. The outermost for-loop iterates n times and each iteration is $\log(n)$, hence the total complexity is $\Theta(n \log(n))$.

b)
```
for i = 1 to n
  for j = 1 to i*i
    for k = 1 to n
      x = x + 1
```

The innermost for-loop iterates n times. The middle for-loop iterates i^2 times depending on the variable i from the outermost for-loop for a run time of $\sum_{i=1}^{n} i^2 n$.

$$\sum_{i=1}^{n} i^2 n = n \sum_{i=1}^{n} i^2 \leq n \sum_{i=1}^{n} n^2 = \sum_{i=1}^{n} n^3 = n^4 \in O(n^4)$$

and

$$\sum_{i=1}^{n} i^2 n = n \sum_{i=1}^{n} i^2 \geq n \sum_{i=\frac{n}{2}}^{n} i^2 \geq n \sum_{i=\frac{n}{2}}^{n} (\frac{n}{2})^2 = \sum_{i=\frac{n}{2}}^{n} \frac{n^3}{4} = \frac{n^4}{8} \in \Omega(n^4)$$

Hence, the total complexity of all three nested loops in $\Theta(n^4)$.

c)
```
i = n
while (i > 1)
  for j = 1 to n
    x = x + 1
    if i is odd
      i = i - 1
    else
      i = i/2
```

The body of inner `for`-loop runs in constant time and changes i. If $i > 0$ then i will be reduced by at least 1 for each iteration. If i is odd it will be reduced by exactly 1. If i is even it will be reduce by $\frac{i}{2}$ which is greater that or equal to 1 when $i \geq 2$. Since the `for`-loop runs n times, the value of i will be reduced from n to 0. Hence the `while`-loop will execute only once. The inner `for`-loop iterates n times and the `if`-statement is constant time for a total complexity in $\Theta(n)$.

d)
```
i = 2
while (i < n)
  for j = 1 to n
    x = x + 1
  i = i*i
```

The inner `for`-loop runs n times and the body runs in constant time. The outer `while`-loop will run until $i \geq n$. Since i is squared each time, let k be the number of times i (with the initial value 2) is squared before i is no smaller than n for the first time. Hence we have $2^{2^k} \geq n$, which is equivalent to $k \geq \log\log n$. On the other hand, we had $2^{2^{k-1}} < n$, which is equivalent to $k < \log\log n + 1$. Hence $k = \lfloor \log\log n \rfloor$ and the run-time is in $\Theta(kn) = \Theta(n \log\log(n))$.

Question 5 [4 marks]

$$f(n) = \begin{cases} n^2 & \text{if } n \text{ is even,} \\ 1 & \text{if } n \text{ is odd.} \end{cases}$$

Since both cases are in $O(n^2)$ then $f(n) \in O(n^2)$.

When n is odd, $f(n) = 1$ and there is no $c > 0$ and $n_0 > 0$ such that $cn^2 \leq 1$ for all $n \geq n_0$ as long as we pick $n > n_0, n > \frac{1}{\sqrt{c}}$ and n odd. So $f(n) \notin \Theta(n^2)$.

When n is even, $f(n) = n^2$ and there is no $n_0 > 0$ such that $n^2 \leq cn^2$ for all $n \geq n_0$ as long as we pick $n > n_0$, $c < 1$ and n even. So $f(n) \notin o(n^2)$.

Note: There are many possible solutions, e.g. functions where $f(n)$ alternates between $o(g(n))$ and $\Theta(g(n))$ would work here.