University of Waterloo
CS240 Fall 2017
Assignment 1

Due Date: Wednesday, Sept 20, at 5:00pm

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f17/guidelines.pdf` for guidelines on submission. This assignment contains only written questions. Submit your written solutions electronically as a PDF with file name a01wp.pdf using MarkUs. We will also accept individual question files named a01q1w.pdf, a01q2w.pdf, a01q3w.pdf, a01q4w.pdf if you wish to submit questions as you complete them.

Note: you may assume all logarithms are base 2 logarithms: $\log = \log_2$.

Problem 1 [4+4+4+4=16 marks]

Provide a complete proof of the following statements from first principles (i.e., using the original definitions of order notation).

a) $27n^7 + 17n^3 \log n + 2016$ is $O(n^9)$

b) $n^2 (\log n)^{1.0001}$ is $\Omega(n^2)$

c) $\frac{n^2}{n+\log n}$ is $\Theta(n)$

d) n^n is $\omega(n^{20})$

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 $f(n) \in \sqcup(g(n))$. Prove the relationship using any relationship or technique that described in class.

a) $f(n) = n^2 + 27n \log n + 2016$ versus $g(n) = n^2 \log n + 2016$

b) $f(n) = 10^n + 99n^{10}$ versus $g(n) = 75^n + 25n^{27}$

c) $f(n) = \sqrt{n}$ versus $g(n) = (\log n)^7$

Problem 3 [4+4+4+4=16 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) $f(n) \notin o(g(n))$ and $f(n) \notin \omega(g(n)) \Rightarrow f(n) \in \Theta(g(n))$

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

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

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

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

Analyze the following piece of pseudocode and give a tight (Θ) bound on the running time as a function of n. Show your work. A formal proof is not required, but you should justify your answer (in all cases, n is assumed to be a positive integer).

a)
```
x = 0
for i = 1 to n + 15 do
   x = x * 4
   for j = 420 to 22100
      for k = 2i to 3i
         x = x * 42
```

b)
```
x = 0
for i = 1 to ceiling(log(n))
   for j = 1 to i
      for k = 1 to 20
         x = x + 1
```

c)
```
x = 0
for i = 1 to sqr(n)    \\ i.e. n^2
   for j = 1 to ceiling(log(i))
      x = x + 1
```