University of Waterloo
CS240 - Fall 2018
Assignment 1

Due Date: Wednesday September 19 at 5:00pm

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f18/guidelines.pdf` for guidelines on submission. All problems 1 – 7 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`, ... , `a01q7w.pdf` if you wish to submit questions as you complete them.

There are 61 marks available; the assignment will be marked out of 60.

Problem 1 [3+3+3+3+3=15 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: $\log = \ln$.

a) $12n^3 + 11n^2 + 10 \in O(n^3)$

b) $12n^3 + 11n^2 + 10 \in \Omega(n^3)$

c) $12n^3 + 11n^2 + 10 \in \Theta(n^3)$

d) $1000n \in o(n \log n)$

e) $n^n \in \omega(n^{20})$

Problem 2 [4+4=8 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) $f(n) = \sqrt{n}$ versus $g(n) = (\log n)^2$

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

Problem 3 [6+6=12 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) $\min(f(n), g(n)) \in \Theta\left(\frac{f(n)g(n)}{f(n)+g(n)}\right)$

Problem 4 [4 marks]

Derive a closed form for the following sum:

$$S(n) = \sum_{i=1}^{n} i/2^i.$$

Problem 5 [2+2+4+4=12 marks]

Consider the following procedure.

```
pre: n is a positive integer
pre: v[1..n] is a binary vector of length n,
     i.e., each entry is either 0 or 1
foo(v,n)
     i := 1;
     while i<=n and v[i]=0 do
         i := i+1
     od;
     for j from 1 to i do
         print("Hello world!")
     od;
```

a) How many inputs are there are of size n?

b) What is the worst case number of calls to print? Give an exact formula in terms of n and justify your answer by giving an example of a worst case input of size n.

c) For $i \in \{1, 2, \ldots, n\}$, let S_i denote the subset of inputs of size n for which the number of calls to print is i. Describe and enumerate S_i.

d) What is the average case number of calls to print? Derive an exact closed form formula in terms of n.

Problem 6 [5 marks]

Prove that the following code fragment will always terminate.

```
s := 3*n  // n is an integer
while (s>0)
   if (s is even)
      s := floor(s/4)
   else
      s := 2*s
```

Problem 7 [5 marks]

Analyze the following piece of pseudo-code and give a tight bound (i.e. Θ notation) on the running time as a function of n. Show your work. A formal proof is not required, but you should justify your answer.

```
mystery ← 0
for i ← 1 to 3n do
    mystery ← mystery × 4
    for j ← 1388 to 2010 do
        for k ← 4i to 6i do
            mystery ← mystery + k
```