University of Waterloo
CS240 Winter 2023
Assignment 1

Due Date: Wednesday, January 25 at 5:00pm

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. You should read and sign the Academic Integrity Declaration before you start working on this assignment and submit it **before the deadline of January 18**; i.e. **read, sign and submit AID01.txt now or as soon as possible** (if you haven't already). The agreement will indicate what you must do to ensure the integrity of your grade. If you are having difficulties with the assignment, course staff are there to help (provided it isn't last minute).

The Academic Integrity Declaration must be signed and submitted on time or the assessment will not be marked.

Please read `https://student.cs.uwaterloo.ca/~cs240/w23/assignments.phtml#guidelines` for guidelines on submission. **Each question must be submitted individually to MarkUs as a PDF** with the corresponding file names: a1q1.pdf, a1q2.pdf, ... , a1q6.pdf . It is a good idea to submit questions as you go so you aren't trying to create several PDF files at the last minute.

Late Policy: Assignments are due at 5:00pm, with the grace period until 11:59pm. Assignments submitted after 11:59 on the due date will not be accepted but may be reviewed (by request) for feedback purposes only.

Notes:

- Logarithms are in base 2, if not mentioned otherwise.

Question 1 [3+3+3+4+4=17 marks]

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

a) $7n^3 \log n + 42n^2 - 15n \in O(n^3 \log n)$

b) $5n^5 - 100n^4 \in \Omega(n^5)$

c) $\frac{n^3}{n-10} \in \Theta(n^2)$

d) $25n^4 + 100n^2 \in o(n^5)$

e) $\sqrt{n} f(n) \in \omega(f(n))$, where $f(n)$ is a function taking positive values.

Question 2 [4+4=8 marks]

Prove the following statements based on the definitions of the order notations. All functions take positive values.

a) If $f(n) \in \Theta(g(n))$, then $f(n) + g(n) \in \Theta(f(n))$.

b) If $f(n) \in \Theta(g(n)), h(n) \in \Theta(r(n))$, and $g(n) \in o(r(n))$ then $f(n) \in o(h(n))$.

Question 3 [3+3+3+3=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 described in class. You can also use the properties from Question 2, even if you did not prove them.

a) $f(n) = n^3 + 2n^2(\log n) + 13n$ versus $g(n) = n^3 \log n + 14n$

b) $f(n) = (2 + (-1)^{\lceil n \rceil})n^2$ versus $g(n) = 100n \log n + 20n$

c) $f(n) = 4n^3 \log n$ versus $g(n) = 8^{\log n}$

d) $f(n) = 2^{2^{n+1}}$ versus $g(n) = 2^{2^n}$.

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

Analyze the following pieces 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
   x = x + 1
   j = i
   while j < n
     j = j + 1
     x = x + 1
```

b)

```
x = 0
for i = 1 to ceiling(log(n))
   for j = 1 to i
      for k = 1 to 10
         x = x + 1
```

c)
```
s = 0;
for i = 1 to n*n do
   for j = 1 to i*i do
        s = s + 1;
```

d)
```
i = 2
x = 0
while (i < n)
  for j = 1 to n
    x = x + 1
  i = i * i * i
```

Question 5 [4+4 = 8 marks]

True or false? For each of the following assertions, indicate whether it is true or false. If true, prove it; if false, give a counter-example and briefly justify it. In both cases, f and g are functions that take positive values.

a) If $f(n)$ is in $\Theta(g(n))$, then $f(n)^2$ is in $\Theta(g(n)^2)$.

b) If $f(n)$ is in $\Theta(g(n))$, then $f(n)^n$ is in $\Theta(g(n)^n)$.

Question 6 [3 marks (+3 bonus)]

For n a power of two, let $T(n)$ be the number of times the following method prints x. Give a recurrence for T. Bonus: solve the recurrence equation to get a closed form.

```
foo(A[1..n]) # assume n>=1 is a power of two
   print("x")
   if n>1 then
      k = 1
      while k<n do
          k = 2*k
          foo(A[1..n/k])
```