University of Waterloo
CS240 Spring 2025
Assignment 1

Due Date: Tuesday, May 20 at 5:00pm

Read `https://student.cs.uwaterloo.ca/~cs240/s25/assignments.phtml#guidelines` for guidelines on submission. **Each question must be submitted individually to Crowdmark.** Submit early and often.

Grace period: submissions made before 19:59PM on May 20 will be accepted without penalty. Your last submission will be graded. Please note that submissions made after 19:59PM **will not be graded** and may only be reviewed for feedback.

Reminder: all logarithms are in base 2 unless stated otherwise.

1. [3+3+4+4] Provide a complete proof of the following statements from first principles (i.e., using the original definitions of order notation).

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

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

   (c) $n \log(\log(n)) \in o(n(\log(\log(n)))^2)$

   (d) $n^n \in \omega(n^{n/2})$

2. [10 marks] For each of the following pairs of functions $f(n)$ and $g(n)$, determine the "most appropriate" symbol in the set $\{o, \omega, \Theta, O, \Omega\}$ to complete the statement that $f(n) \in \quad (g(n))$ (if one of the symbols applies at all). "Most appropriate" means that you should not answer "O" if you could answer "o" or "Θ". Justify your answers even in the cases when none of symbols applies at all.

   (a) $f(n) = 2025n^3 + 12871n^2 + 19$, $g(n) = \frac{2}{2025}n^4 + 2n$;

   (b) $f(n) = \log^2(n^4)$, $g(n) = \sqrt{n}$;

   (c) $f(n) = 16^{\log n^3} + n^5, \quad g(n) = \frac{1}{2}n^{12} + 100n^6$;

   (d) $f(n) = n^2$, $g(n) = (\lceil \frac{n}{2} \rceil - \frac{n}{2})n^2$;

3. [6 marks] Prove or disprove each of the following statements (give a counter example to disprove).

   (a) If $\log(f(n)) \in \Omega(\log(g(n)))$ then $f(n) \in \Omega(g(n))$.

   (b) If $f(n) \in O(h(n))$ and $g(n) \in O(h(n))$ then $\dfrac{f(n)}{g(n)} \in O(1)$.

(c) If $f(n) \in o(g(n))$ then $\log(f(n)) \in o(\log(g(n))$.

4. Analyze the following pseudocode and give a tight (Θ) bound on the running time as a function of n. You can assume that all individual instructions (including logarithm) are elementary, i.e. take constant time. Show your work.

   (a) [3 marks]

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

   (b) [3 marks]

   ```
   l = 0
   for i = n + 1 to n^2 do
      for j = 1 to ceiling(log(i)) do
          l = l + 1
      od
   od
   ```

   (c) [8 marks]

   ```
   m = 1
   l = 0
   s = 0
   while m <= n do
      for j = n - m to n do
         l = l + 1
      od
      for j = 1 to ceiling(log n) do
         s = s + 1
      od
      m = 3 * m
   od
   ```

5. [2+6 marks]

   We consider two algorithms, Algo1 and Algo2, that solve the same problem. For any input of size n, Algo1 takes time $T_1(n)$ and Algo2 takes time $T_2(n)$. 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, provide a counter example and explain it.

   (a) Suppose that $T_1(n) \in O(n^2(\log n)^5)$ and $T_2(n) \in O(n^3(\log n)^2)$. Does it imply that there exists n_0 such that for $n \geq n_0$, Algo1 runs faster than Algo2 on inputs of size n?

(b) Same question, assuming that $T_1(n) \in \Theta(n^2)$ and $T_2(n) \in \Theta(n^3)$.