

University of Waterloo

CS240 Spring 2026

Assignment 1

Due Date: Tuesday, May 26 at 5:00pm

Read <https://student.cs.uwaterloo.ca/~cs240/s26/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 26 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.

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

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

- a) $5n^3 + n^2 + 10n + 7 \in O(n^3)$
- b) $n^2(\log n)^{0.00001} \in \Omega(n^2)$
- c) $\frac{n^2}{n+\log n} \in \Theta(n)$
- d) $n \log(\log(n)) \in o(n(\log(\log(n)))^2)$
- e) $2^n \in \omega(n^{1/n})$

Question 2 [3+3+3+3 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 + 12 do
  x = x * 4
  for j = 389 to 20100
    for k = 2i to 3i
      x = x * 77
```

- 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) `s = n`
`while (s > 0)`
`if (s is even)`
`s = floor(s / 4)`
`else`
`s = 2 * s`

Question 3 [2+2+2 marks]

You are a TA for CS240 during W27. It is a cold snowy morning in Waterloo. You wake up late, spend 30 minutes shoveling snow, and barely make it to your office hours on time. Unfortunately, you did not even get a chance to buy coffee. You sit down tired, frozen, and questioning all of your life choices.

A few minutes later, a cheerful student named Leo walks in holding a large cup of coffee and asks for help proving that the recurrence

$$T(n) = T(n - 1) + 1, \quad T(1) = 1$$

has linear runtime.

Leo says he is trying to prove

$$T(n) \leq cn$$

for some constant c , using induction.

He explains his proof:

Assume

$$T(n) \leq cn.$$

We show that

$$T(n + 1) \leq c(n + 1).$$

Since

$$T(n + 1) = T(n) + 1,$$

using the induction hypothesis,

$$T(n + 1) \leq cn + 1.$$

Leo then writes

$$cn + 1 \leq cn + n = (c + 1)n.$$

Therefore,

$$T(n + 1) \leq (c + 1)n.$$

Leo smiles confidently, takes a sip of coffee, and says:

“This is still linear, so the proof should work!”

As the TA, you are not convinced.

- a) Assume Leo continues using the same style of argument repeatedly. Determine what upper bound he would get for

$$T(n + 2), T(n + 3), \dots, T(2n).$$

- b) Explain why this shows that allowing the constant to increase at each induction step can eventually lead to a non-linear bound here. Explain why this is not a valid way to prove a linear upper bound in general.
- c) Tell Leo what a correct induction proof would look like here.

Your answer should clearly explain why the inductive step must end with the same constant c as the induction hypothesis.

Strangely, by the end of the conversation, Leo’s excitement about the problem and the fun of thinking carefully about the proof have worked better than coffee. For a moment, the snowy morning no longer feels so terrible, and you remember that you are actually happy with most of your life choices, including studying computer science at UWaterloo.

The following exercises are provided for additional practice and are not part of the graded portion of the assignment. While these questions will not be evaluated, we strongly encourage you, and expect you, to work through them to deepen your understanding of the material. You are welcome to upload your solutions to Crowdmark for your own record-keeping and organization; however, no grades or feedback will be provided for these submissions. You may also ask questions about these exercises during office hours or on Piazza.

Question 4

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.

1. $f(n) = 2025n^3 + 12871n^2 + 19$, $g(n) = \frac{2}{2025}n^4 + 2n$;
2. $f(n) = \log^2(n^4)$, $g(n) = \sqrt{n}$;
3. $f(n) = 16^{\log n^3} + n^5$, $g(n) = \frac{1}{2}n^{12} + 100n^6$;
4. $f(n) = n^2$, $g(n) = (\lceil \frac{n}{2} \rceil - \frac{n}{2})n^2$;

Question 5

Prove or disprove each of the following statements (give a counter example to disprove).

1. If $\log(f(n)) \in \Omega(\log(g(n)))$ then $f(n) \in \Omega(g(n))$.
2. If $f(n) \in O(h(n))$ and $g(n) \in O(h(n))$ then $\frac{f(n)}{g(n)} \in O(1)$.
3. If $f(n) \in o(g(n))$ then $\log(f(n)) \in o(\log(g(n)))$.

Question 6

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.

1. 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 ?
2. Same question, assuming that $T_1(n) \in \Theta(n^2)$ and $T_2(n) \in \Theta(n^3)$.