

University of Waterloo

CS240 Spring 2022

Assignment 1

Due Date: Wednesday, May 18 at 5:00pm

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. As part of every assessment in this course you must read and sign an Academic Integrity Declaration before you start working on the assessment and submit it **before the deadline of May 18** along with your answers to the assignment; i.e. **read, sign and submit A01-AID.txt now or as soon as possible**. 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/s22/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, ... , a1q5.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 on Wednesday. Students are allowed to submit **one** late assignment, 2 days after the due date on Friday by 5:00pm. Assignments submitted after Friday at 5:00pm or Wednesday at 5:00pm (if you have already used your one late submission) will not be accepted for grading but may be reviewed (by request) for feedback purposes only.

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

Question 1 [3+3+4=10 marks]

Provide a complete proof of the following statements from first principles (i.e., using the original definitions of order notation) if the statement is true. Otherwise simply state that it is false.

a) $7n^4 - 5n^2 + 6 \in O(n^4)$

b) $7n^4 - 5n^2 + 6 \in \Omega(n^4)$

c) $5n^2 + 15 \in o(n^3)$

Question 2 [3+3+4=10 marks]

For each pair of the following functions, fill in the correct asymptotic notation among Θ , o , and ω in the statement $f(n) \in \square(g(n))$. Provide a brief justification of your answers. In your justification you may use any relationship or technique that is described in class.

- a) $f(n) = n^2 + 22n(\log n) + 13$ versus $g(n) = n^2 \log n + 14$
- b) $f(n) = \sqrt{n}$ versus $g(n) = (\log n^4)$
- c) $f(n) = 10^n + 99n^{10}$ versus $g(n) = 75^n + 25n^{27}$

Question 3 [4+4+4+4+4=20 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 show that the truth of the statement leads to a contradiction. All functions map positive integers to positive integers.

- a) If $f(n) \in O(g(n))$ then $g(n) \in \Omega(f(n))$.
- b) There exists $f(n)$ and $g(n)$ such that $f(n) \in o(g(n))$ and $f(n) \in \omega(g(n))$.
- c) If $f(n) \in O(g(n))$ then $2^{f(n)} \in O(2^{g(n)})$.
- d) $(\log n)^{\log n} \in O(n^2)$.
- e) $\log n \times 2^{\sin(n^3)} \in O(n)$.

Question 4 [2+4=6 marks]

Dr. I. M. Smart has recently invented a new class of functions, denoted $O'(f)$. A function $g(n)$ is in $O'(f)$ if there is a constant $c > 0$ such that $g(n) \leq cf(n)$ for all $n > 0$. Prove the following statements about the relationship between $O(f)$ and $O'(f)$.

$f(n)$ and $g(n)$ are functions that map positive integers to positive integers.

- a) Prove that $f(n) \in O'(g(n))$ implies that $f(n) \in O(g(n))$.
- b) Prove that $g(n) \in O(f(n))$ implies that $g(n) \in O'(f(n))$.

Question 5 [4+4+5=13 marks]

For the questions below, analyze the following pieces of pseudocode and for each of them give a tight (Θ) bound on the running time as a function of n . Your proof should be as formal as possible (similar to the analysis performed in class). In all cases, n is a positive integer.

```
a) for i := 1 to n do
    for j := 1 to i * i do
        for k := 1 to log(n)
            s := s + 1
```

```
b) s = 1
   i = 2
   while (i < n) do
       for j = 1 to n do
           s = s + 1
       i = i * i
```

```
c) s := -42
   i := 1
   while i < 5n do
       j := n * n * n
       while j > i do
           s := 2 - s
           j := j - i
       i := i + 5
```