# University of Waterloo
# CS240E, Winter 2025
# Written Assignment 3 Post-Mortem

This document goes over common errors and general performance on the assignment. We create it using feedback from the graders, and it is meant to be used as a resource to understand common areas that we can improve in.

## Question 1 [5+7=12 marks]

- This question was done well.

## Question 2 [5 marks]

- Many solutions did not argue that the bound they obtained is tight. To clarify, "tight" does not mean "exact" as opposed to "asympotic", it means "least possible" for an upper bound (or "greatest possible" for a lower bound).
- Proving tightness is especially important here because many of these solutions ignored rounding in the induction, which could give a bound that is not tight, since $\log \log n$ grows so slowly.

## Question 3 [3 marks]

- Most solutions were correct, but some forgot to explicitly state that the optimal static order was decreasing (i.e. the order they were originally given in).

## Question 4 [2+2+2+5+4+2(+5)=17(+5) marks]

- Most of parts (a)-(f) were done well, although some solutions miscalculated a tree's height by 1. Always remember that the height is one less than the number of layers.
- Most attempted solutions to the bonus part (g) had something close to the correct choice of $f$ (multiple choices of $f$ could be correct depending on the divisibility assumptions), but some of them did not clearly show that it was both an upper and lower bound.

## Question 5 [2+6=8 marks]

- Part (a) was done well.
- In part (b), many solutions considered several cases and subcases, and occasional got some details wrong within them or made unnecessary assumptions on $n$. In such cases it may have been easier to just consider whether $k$ is in the array or not.

## Question 6 [5 marks]

- Some solutions only found the length of the longest prefix, but the questions asks for an algorithm that actually returns a prefix.
- Some solutions also used depth-first search without further elaboration, which (despite its ubiquity in competitive programming) is not something we've covered in this class, and hence requires explanation or pseudocode.