# University of Waterloo
# CS240 Spring 2023
# Assignment 2 Post-Mortem

This document goes over common errors and general student performance on the assignment questions. We put this together using feedback from the graders once they are done marking. It is meant to be used as a resource to understand what we look at while marking and some common areas where students can improve in.

## General

- Some of the students have missed justification of correctness for their algorithm. If your pseudocode is based on algorithms given in lecture, one line saying "correctness follows from `fix-up`, for example, from lecture" is good enough for most of questions in this assignment.
- Please make sure that your work is nice and clear for the reader to follow. Poor presentation (illegible handwriting, scanning not done clearly) may lead to deduction.

### Question 1 [10 marks]

- Overall, this question was done very well.

### Question 2 [5 marks]

- Some of the students have only included `fix-down` or `fix-up`. Which function to call depends on how the heap property is violated after switching `A[i]` and `A[n-1]`. Another trick to go around this is to call both. If it is a case when `fix-down` is required, calling `fix-up` on that heap is not going to do anything.

### Question 3 [1+3+4+5 marks]

- When implementing `delete` operations, some students did not modify another heap. Foe example, within `deleteMin`, you should modify not only $H_1$, but also $H_2$. To modify $H_2$, you could use your alogorithm from Question 2.
- Before deleting an item from one heap, you should store the index of item in other heap that will get removed. For example, if we are dealing with `deleteMin`, we should store the index of minimum item in $H_2$ before we remove it and update $T_1, T_2$.

## Question 4 [1+4+5 marks]

- Some students merged arrays one by one. What we mean is that if we have $A_1, A_2, A_3, A_4$, one would call `merge` on $A_1$ and $A_2$ and let resulting one $A_5$. Then, one calls `merge` on $A_5$ and $A_3$. Careful analysis will get more than $ns$ comparisons which we aimed.
- Some students have shown bound of number of comparisons, while the question asked for precise expression.
- Some students have used idea of keeping track of smallest item from each $k$ sub-array when merging $k$ sub-arrays. Linear search through those items will lead to $ns$ comparisons.