University of Waterloo
CS240 Spring 2023
Assignment 3 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 algorithm from lecture" is enough.
- 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 [7 marks]

- We asked students to be explicit about how you represent the data. If one chose to use any type of Radix-Sort or Bucket-Sort for this question, one had to explain how to extract dth digit, since this is important for value comparison.
- Adding leading 0s are one other details that one could take care of. This does not necessarily have to be done if we extract digits using floor and mod.

Question 2 [7 marks]

- Some students did not modify recursive call correctly to use `FastMerge`. Common mistakes that we noticed was to split array by half once, which is similar to what was done in lecture. With this assignment, the key was to modify those recursive calls.
- Some students did not provide full details on what the contradiction was. Correct contradiction should contain an idea that any comparison-based sorting algorithm has to have run-time of $\Omega(n \log n)$. But, should FastMerge exist, the modified MergeSort is in $o(n \log n)$, contradicting the statement earlier.

Question 3 [5+4 marks]

- With part a), some students did not consider using decision tree. Unless one can prove the property across all possible algorithms solving this problem (which is not possible), presenting an algorithm as proof is not valid.

- When doing decision tree proof, it is important to discuss properties of the decision tree in general terms. Particular decision trees arises from particular inputs or particular algorithms – we seek some more general properties exhibited by the decision trees required to solve this problem in general terms.
- Some students missed to apply ceiling on their final expression. It is minor, but yet, important details to be included.
- With part b), some students did not provide justification of correctness. Simple statement like "my algorithm takes care of all possible outcomes while meeting lower bound requirement" is enough.
- If one did part a) correctly, one should only use 2 weighing in their algorithm in part b). Doing more or less than 2 weighing received deduction because it is either incorrect or not efficient.

Question 4 [3+2+4+1+1+3+3 marks]

- With part b), some students stated that *one partition has size of* $n - \sqrt{m}$ *where as one side has size of* $\sqrt{m}$ at the end of partitioning whole array with n elements. This would be true in the worst case specific, however, one should avoid referring to specific scenario here. If you wish to refer to worst case, you would have to explain on why every other case also must have size at least $\sqrt{m}$.
- With part c), some students did not consider the cost of very first insertion and partition. Cost of very first insertion sort can be found by what was given in the question as we know the input size. Partition, based on what we have discussed in the lecture, does $O(m)$ comparisons where m is the number of elements in the array.
- Some students did not refer to part e) for their proof in part f) and g). Referring to part e) provides justification on why it is that you could move from $k-1$th iteration to kth iteration.
- Some students assumed that $T(n)$ is increasing function, which you cannot. That is, if one stated $T(a) < T(b)$ where $a < b$ is given, appropriate deduction had to be made.