University of Waterloo
CS240 Spring 2023
Assignment 4 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]

- Some students built an algorithm that is based on "normal" notation of order (i.e. 1st, 2nd,..., nth). However, it was clear in the question that k is an index value - such mistake received deduction.
- Some students made a mistake in modifying k value when making a recursive call to right subtree.
- Some students missed to check whether current node's child is missing or not. For example, in the sample solution, it explicitly checks whether left child exists or not before accessing number of keys that left child is storing. One should either explicitly check whether left child exists or one should define that if node does not exist, accessing number of keys will return 0.

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

- Part a) and b) were very well done.
- With part c), some students did not mention AVL tree's height property where current node's left and right subtree has different in their height at most 1. That is, during inductive hypothesis, one should mention above property to justify the maximum height of another child.

- With part d), some students referred to lecture and stated $h \in \Theta(logn)$. However, this does not mean that $\log n \leq h$. One should include constant or refer to exact expressions that was introduced during the lecture - even in then case, one should have explained more details in proving inequalities.

Question 3 [4+4 marks]

- This question well done overall.

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

- Part a), b), and c) were well done overall.
- With part d), please refer to piazza post @985 for more details.
- Some students did not prove two inequalities included in the hint and they received deductions accordingly.

Question 5 [5 marks]

- Many students missed analysis on upper bound of interpolation search. As you could find in the sample solution, this is trivial to prove - interpolation search algorithm should do no more than n iterations to locate an element, which presents upper bound of interpolation search algorithm on any input.
- Similar to sample solution, some students gave a formula to generate an array in terms of n and showed that such array takes n iterations. Detailed calculation was required, especially how m value changes in each iteration. Deductions were given if this was missed.
- Some students did not provide such array and simply stated that search space on array A reduces by 1 in each iteration - this is the correct idea, but more work needs to be shown to demonstrate that this is indeed the case at every step, and what input is needed to achieve this.
- Many students gave an array and showed that such array leads to n iteration and directly concluded that worst-case run-time is $\Theta(n)$. However, this only implies that with such input array, interpolation search does n iterations, not that interpolation search has to do n iterations at most for *any* inputs. The general claim, then, is a necessary claim that needed to be substantiated if an example is provided as justification. However, as mentioned, this is trivial to prove that no matter what elements you have, interpolation search algorithm should be able to terminate after looking all items, once each.

Question 6 [5 marks]

- This question was well done.