University of Waterloo
CS240R Spring 2017
Assignment 1 Post mortem

Written Friday, May 26

Problem 1

Students generally picked the correct c and n_0 for parts a) and b) but struggled with c) and d). Part d) was poorly done, with many students trying to isolate for n in the inequality, realizing that it was impossible, and then expressing n_0 in terms of n, which does not make sense. n_0 must be expressed in terms of c.

Students who did d) correctly realized that they had to express n_0 as a function of c such that the inequality held true for all possible $c > 0$. Many correct solutions were given by these students.

Part c) had a few students stumble on their choice of c_1 and c_2. Some made the mistake of using a larger c_1 despite that it is the constant for the lower bound of $f(n)$.

Problem 2

While part a) was mostly done well, students should be aware that simply writing down the definitions of little-o and little-ω is not enough. Some students assumed that n_0 was equal for both order notations, which is not true.

Parts b) and c) were done poorly. Students used limits for b) despite the question stating that answers had to be based on the definitions of each order notation. Students often wrote long, complex, and incorrect proofs for b) when a much simpler proof starting from $log(n) \in o(n^i) \forall i > 0$ would have been sufficient.

Part c) had many students assuming that the statement was true. While the summation is indeed upper bounded by n^k, it is not lower bounded by that.

Some students also tried to take $n = 1$ as a counterexample. Order notation is concerned with the growth rates of functions and therefore looks at n as it grows very large, so taking an arbitrarily small value of n does not suffice to be a proof.

Problem 3

This problem was generally done well by most students. A common error was writing only a big-O bound for the runtime of the algorithm instead of a Θ bound. Some students also had problems evaluating the summation into a formula. In part c), a few students arrived at a $\Theta(n)$ bound instead of a $\Theta(2^n)$ bound.

Problem 4

This was done well by most students.

Problem 5

Generally done well, but with some recurring errors. In part a), some students did not provide the correct formula for finding the parent or children of a node i. In part b), a few students did not start with the right bound for n, and thus ended up with an upper bound for h that was different than the one asked in the question. The question asked for what the height of a ternary heap of n nodes can be at most, not what the height can be with at most n nodes. Filling up the last level of the heap does not increase the height at all.

A few students did the bonus question correctly. Many students who attempted the bonus failed to provide an algorithm that ran in $O(log(n))$ time and justified it incorrectly.