A lot of students wrote too much justification for correctness of their algorithms, and went on for paragraphs giving a thorough proof. A proof is only required if the question asks to “argue” or “prove” something. When doing algorithm design, usually only a brief justification is necessary. It needs to be concise and clear, demonstrate that you understand the answer and convince the TAs that your answer is correct. Same goes for running time analysis.

Students are highly encouraged to look at sample solutions and thoroughly read that assignment guidelines:

No marks were taken off for writing answers that were too long (presuming they were coherent). However, on future assignments, if answers are really excessively long, we will be tempted take marks off. So try to keep your level of detail to no more than what was on assignment solutions. (Quite frequently even the assignment solutions are more detailed than what would have been needed.)

**Question 1b)**

Many students forgot that heapify takes linear time. Some students didn’t explain how to do the sorting using Prof Diot’s algorithm.

**Question 3**

Almost everyone did not explain how to convert decimal numbers to base n numbers (or to whatever base they were using for this question for radix sort).

**Question 4d)**

This is one question which actually asked to “argue” the running time, which means a more formal argument was required. Surprisingly, (given how much detail the other questions were done with), most students did not have enough detail for this argument.

**Question 4f)**

Nobody got the most efficient (HARD) solution with $O(n \log k)$ running time and $O(1)$ auxiliary space. Many got close. Thus no one got a 10 in the bonus part. Many students
did well to get a solution with $O(n \log k)$ running time and $O(\log k)$ space for which they got a maximum of 6/10.