University of Waterloo
CS240 Spring 2022
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.

- In general, many student answers were lacking in correctness justification for algorithm design questions. Simply explaining how the algorithm works is not sufficient. Your justification should indicate why your solution is always guaranteed to produce correct output.

Question 1 [4 marks]

- Many students forgot to use `fix-up` in their algorithm for when the swapped key (from the last node in the heap) is larger than its new parent's key.

Assignment Project Exam Help

- Some students deleted the element before swapping it with the last node in the heap.

https://tutorcs.com

Question 2 [8 marks]

- Some students did not explicitly state the changes needed to make to $MergeSort$ so that merging arrays can be done with $FastMerge$ instead of $Merge$.

WeChat: cstutorcs

- Some students gave a running time in terms of m but did not establish a relationship between m and n (i.e. $m^2 = n$ or $m = \sqrt{(n)}$). Since the question specified the array's size to be n, the running time must be related to n.

- Some students incorrectly stated that the lower bound on comparison-based sorting algorithms is $O(n \log n)$ instead of $\Omega(n \log n)$. $O-$notation implies an **upper** bound and should not be used in a lower bound analysis.

- Although many students mentioned that the lower bound on comparison-based sorting algorithms is $\Omega(n \log n)$, some did not mention that this applies to **comparison-based sorting algorithms**. Since non-comparison based sorting algorithms can run in $o(n \log n)$ time, this is an important distinction to make.

- Some students did not state the actual contradictory statement (i.e. $n \log \log n \in o(n \log n)$).

Question 3 [8 marks]

- Some students mentioned the number of blocks they intend to divide the array into but not the size of each block (and vice versa).
- Some students used the term "consecutive" to refer to blocks separated by a single block, rather than to adjacent blocks and then incorrectly stated that consecutive sorted blocks can be concatenated/merged together. This was overlooked if any pseudocode was provided that indicated the student actually meant blocks separated by a single block.
- Some students did not realize that after sorting the $\frac{n}{\log n}$ sub-arrays $A_1, A_2, \ldots, A_{\lfloor \frac{n}{\log n} \rfloor}$, simply concatenating the sub-arrays at odd positions and the sub-arrays at even positions would result in 2 sorted arrays $[A_1|A_3|A_5|\ldots]$ and $[A_2|A_4|A_6|\ldots]$.
- Some students divided the array into $\frac{n}{\log n}$ blocks, sorted each block individually, and then merged consecutive blocks together. This does not take advantage of the property that $A[i] \geq A[i-j]$ for all $j > \log n$.
- Some students forgot to merge all the sub-arrays into one final array.
- A few students forgot to justify the time complexity of their algorithm.

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

Question 4 [3+3+4 marks]

- For part (c), some students forgot to mention the type of heap H_{hi} should be.
- For part (c), some students inserted the element into the heap of smaller size, regardless of how the element compares to the current median or the root of the heap.
- For part (c), some students forgot to maintain the size requirement of the two heaps (i.e. H_{lo} has $\lceil \frac{n}{2} \rceil$ elements and H_{hi} has $\lfloor \frac{n}{2} \rfloor$ elements, where n is the total number of elements in the two heaps).
- For part (c), some students did not justify that their running time is in $o(n)$.
- For part (c), some students confused $o(n)$ with $O(n)$. Since $g(n) \in O(f(n)) \not\Rightarrow g(n) \in o(f(n))$, proving that the running time is in $O(n)$ does not suffice here.