

Drill problems — both of the open addressing with sequential probing questions were meant to use the reinsertion strategy to handle deletions. This was only stated for one of the problems.

These are design problems, not research problems — use what has been discussed in class and/or in the readings to design the data structures or algorithms called for rather than looking up a specialized data structure or algorithm.

Also, the exercises are grouped into subsections — those subsection titles can give a big hint about how to solve the problem...

Examples can be valuable for helping to explain things that are tricky to express solely in words, but only giving a couple of specific examples is not a substitute for an algorithm — an algorithm must describe a procedure that works for all valid inputs, while a sequence of example steps only shows what happens in those particular situations.

Examples also need to be tied to a general rule or step rather than existing in isolation — they need to be examples *of something*, and you need to make the “of what” part clear to your reader. For example, just showing before and after pictures of the contents of an array conveys that something new was added to the array but the reader has no idea why the new element went where it did and so could not perform the step in another situation. However, if the pictures are connected to the step “insert the element in sorted order”, the reader is equipped with a general rule they can apply in other cases (insert in sorted order) and the before and after pictures serve to illustrate that step applying in a specific case.

For #3 (ADM 4-7), be careful to avoid off-by-one errors. Are you indexing starting with 0 or 1? This matters if you are comparing indexes and counts to find a specific  $h$  value.

Both bubbling up and bubbling down are used with heaps, but for a particular operation and a particular heap organization (min-heap or max-heap), whether to bubble up or bubble down can be definitively determined. Saying that deleting from a heap involves bubbling up or down is not accurate. Deleting from a heap usually involves swapping values and the bubbling is to fix the newly swapped element — think about the heap-order property and where that element came from in the heap to determine if it is bigger or smaller than the element it replaced, and consider the heap-order property again to determine whether it needs to bubble up or bubble down.

Show your work — for #6 (ADM 4-41), also write the expressions you evaluated to get the expected number of steps. How did you compute the numbers?