HW 2

- #1
  - be sure to deal with special cases (`todel` first node, empty list) and preconditions (`null` value for `todel`)
  - what about `todel` being the last node?

> [3] Actually, there *is* a way to delete an element from a singly linked list in constant time, as shown in Figure 3.2. Overwrite the node that x points to with the contents of what `x.next` points to, then deallocate the node that `x.next` originally pointed to. Special care must be taken if x is the first node in the list, or the last node (by employing a permanent sentinel element that is always the last node in the list). But this would prevent us from having constant-time minimum/maximum operations, because we no longer have time to find new extreme elements after deletion.

HW 2

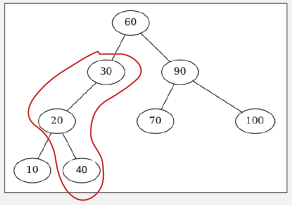
- #2-3 – appropriate level of detail

> *For the "design a data structure" problems (ADM 3-4 and 3-8), use the target runtimes stated in the problem to steer you towards a solution — keep in mind what you know about the running time of standard ADT, array, and linked list operations as well as the tactics discussed in class for improving runtimes (e.g. store instead of search). Your writeup should identify what is stored and how, outline how each operation is carried out, and explain how the desired running time is achieved. Key here is an appropriate level of abstraction — the goal is to express the algorithm with sufficient detail to be understood and to be able to assess its correctness and running time, but not to overwhelm the ideas with details that obscure understanding. Review (and revise) your writeup with an eye towards achieving the right level of detail.*

  - both actual code and highly detailed pseudocode are rarely appropriate, especially without a high-level overview of the core ideas
  - if you can use a known ADT as is, describe your algorithm in terms of those operations, not the internals of how to implement that ADT
    - e.g. if you are using a stack, no need to address array vs linked list implementation and how they work – use push, pop, top

HW 2

- #4, #5 – treat as separate operations, not a sequence
- #4
  - BST isn't a set – not an error to insert a duplicate element
- #5
  - problem didn't actually contain a valid BST – this has been corrected in the posted homework
  - no credit lost if treated as if valid ("OK") but if you redo #5, use the corrected version



HW 2

- #6, #7
  - show the tree after each insert/remove, not only the final result
  - if you show intermediate steps (e.g. after insert, then after restructure), make it clear which trees are the end result of each insert/remove

- **Homework** and **programming assignments** are *individual* assignments intended primarily for practice and learning. You may use resources other than the textbook and slides/ examples posted on the course webpage for learning about a topic, and you may get help from or discuss ideas with others. However, you may not work collaboratively with others to produce a result and you may not do things which shortcut the learning process.

When it comes to figuring out how to do an assignment, your first source of information should be the course materials (textbook, slides, and other materials made available on the course website) and reference materials linked on the course website. For help, you should turn first to office hours. Using other resources, where permitted, can easily cross the line from acceptable to not. **When completing assignments, the use of generative AI systems such as ChatGPT or Codex, "homework help" or "study aid" sites such as Chegg or Course Hero, and sites where you post a homework problem or question and solicit answers from others is very likely to cross the line into shortcutting learning and should be avoided.** The use of ChatGPT to learn about a topic outside of the specific context of a particular task is not forbidden, though it is also not recommended in general since AI can be unreliable in its answers.

- Homeworks and programming assignments: Graded on a 10-point scale based on effort and achievement:
  - 0-1 point — generative AI and/or other resources used as a learning cheat

  Most assignments that earn 3 or more points on the initial handin can be revised and resubmitted once. The grade for the revised version will replace the original grade and, because understanding and correcting mistakes is a valuable part of learning, an additional point will be earned for a substantive revision effort.

- if a solution was flagged with “concern about the source”, see me in order to revise and resubmit