This homework covers dictionary data structures (balanced search trees and hashtables). It also revisits some data structures and applications of data structures previously covered. It is due in class Wednesday, February 18.

See the Policies page on the course website for information about revise-and-resubmit, late work, and academic integrity as it applies to homework.

Write your solutions carefully — your work should be neat, readable, and organized.

For "design a data structure" problems, describe what is stored and how and explain how each operation is carried out. For both data structure and algorithm design problems, if a target running time is given, also explain how your solution achieves it.

Hints for design problems:

- *Use the highest level of abstraction you can — in this case, the heading of the section containing the problem gives a big hint as to whether to be starting with ADTs or concrete data structures in thinking about the solution. Don't jump straight to arrays or linked lists when you could use a stack or queue, for example.*
- *Use target runtimes (when given) to help 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).*

When writing up your solution, keep in mind that the goal is to express the algorithm/data structure idea and operations with sufficient detail to be understood and to be able to assess correctness and running time, but not to overwhelm the ideas with details that obscure understanding.

- *Give a brief overview of the idea before the details. This provides context and is especially important if the process is complex or requires a detailed algorithm.*
- *Avoid unnecessary detail — use the highest level of abstraction you can, don't explain known things, and don't provide implementation specifics if they don't affect the running time. For example, the Queue ADT is well-known so you don't need to explain how to implement enqueue or dequeue (just how those operations are used in your solution), and it is well-known that there is an implementation of Queue where the operations are* $\Theta(1)$ *so you don't need to identify whether the queue uses an array or a linked list. What is known depends on your audience, of course — write for your classmates, so standard things that have been discussed in class or in the reading do not need to be explained in your writeup.*

- *There is something of an art to good writeups and it takes practice — review (and revise) your writeup with an eye towards achieving the right level of detail.*

1. Do the homework #3 drill problems on Canvas. (Look for `hw3 drill` in the Quizzes section.)
2. ADM 3-18, page 104. Also explain how to modify the operations that need to be modified. (Don't only identify them.)
3. ADM 3-25, page 104.
4. ADM 3-11, page 104.
5. ADM 3-2, page 103. (First make sure you understand the task — why is the answer 12 for the example given?)