This assignment covers material from the lectures on Chapters 8, 9, 11, 12, and 13, in preparation for Quiz 4.

Not in textbook: In the search problem, the input is an array $A$ of size $n$ along with a search key $k$. The output is an integer $i$ such that $A[i] = k$, or $-1$ if $k$ is not in $A$. Prove, using the decision tree method, that any correct algorithm for this problem based on comparisons ($<$, $>$, $\leq$, $\geq$, and $=$) between elements takes $\Omega(lg n)$ time.

Pages 219–220: Exercise 9.2-3, 9.2-4
Page 223: Exercise 9.3-3
Page 224: Problems 9.1 (express the run time of each solution in terms of both $n$ and $i$), 9.2b, 9.2c
Page 261: Exercises 11.2-2, 11.2-5, 11.2-6
Pages 268–269: Exercise 11.3-1

Not in textbook: Write pseudocode for an algorithm that uses a hash table to solve the element uniqueness problem:

- Input: An array $A$ of $n$ elements.
- Output: “True” if the elements of $A$ are all distinct, or “False” if $A$ contains at least one pair of duplicate elements.

How efficient is your algorithm in the worst case? How efficient is it under the simple uniform hashing assumption? Can you design a different algorithm, not based on hashing, that performs better?

Page 289: Exercise 12.1-1
Page 293: Exercises 12.2-1, 12.2-4
Page 314: Exercise 13.2.3, 13.2-4

Not in book: Figure 13.10 shows a treap variant called a min-treap, which differs from the version in the notes only because the priorities from a min-heap rather than a max-heap. Show the result of inserting the key ‘J’, with priority 8, into the min-treap shown in Figure 13.10f. Then show the result of inserting the key ‘J’, with priority 1, into the min-treap shown in Figure 13.10f. In each case, list the rotations performed by the insertion.