Dynamic Programming

- Read section 8.1 (pages 283-289)
- What type of problems is **dynamic programming** good for?
  - How is computing the Fibonacci numbers an example of this type of problem?
  - Understand the relation between finding a recurrence relation to a problem and solving it via dynamic programming.
  - What are the *bottom-up* and *top-down* approaches to solving a dynamic programming problem?
- How can we solve the Fibonacci numbers problem with dynamic programming?
  - What is the time efficiency of this implementation?
  - How does this efficiency compare to the time efficiency of the "naive" implementation?
- Describe the **coin-row** problem (example 1)
  - What is the recurrence relation that provides a solution to this problem?
  - Write an algorithm for solving the coin-row problem via dynamic programming.
  - How can we arrange for the algorithm to also keep track of which coins we use and which we do not?
  - Practice: 8.1.2, 8.1.3
- Describe the **change-making** problem (example 2)
  - What is the recurrence relation that provides a solution to this problem?
  - Write an algorithm for solving the change-making problem via dynamic programming.
  - Look at the *ChangeMaking* algorithm on page 287.
    - What is the role of the **temp** variable?
    - What do the two checks in the **while** loop accomplish?
- Practice: 8.1.10, 8.1.11