

---

## Assignment 4 – Due Tuesday, November 1

These are all written (non-programming) questions. Your answers are to be turned in on paper at the beginning of class on the due date. On all problems, *show your work* (describe how you arrived at the answer — don't just give the final answer).

1. Page 835, Exercise 21.3
2. Page 836, Exercise 21.14
3. Page 836, Exercise 21.18
4. Page 342, Exercise 7.20
5. Draw a table and fill it in for the dynamic programming change making algorithm from Section 7.6 (code given in Figure 7.25), showing the minimum number of coins required to make change for up to 32 cents. Do this by hand, not by writing and running a program!
6. Page 343, Exercise 7.27
7. Draw a table and fill it in according to your algorithm from the previous problem, showing the number of ways of making change (with standard US coin denominations) for all values up to 32 cents.
8. Consider this problem: You need to plan a walk down an aisle that passes  $n$  stations, where each station contains an item labeled by its weight, and you can either pick up the item and take it with you, or leave it there. You want to reach the end of the aisle carrying some target weight (like 1 kg). At each station you have to make a choice (pick up the item or not), and if you make the wrong choice you may not discover it until you reach the end of the aisle (or when you're carrying too much) – when you discover your error, you will need to back up and try the other choice, much like trying to find your way through a maze. This is a classic backtracking approach, and the code solving this problem using backtracking is in Bitbucket.
  - (a) Analyze the worst-case running time for this algorithm (you will need to derive a recurrence relation and solve it).
  - (b) On my workstation, a worst-case input (a list of weights for which there is no solution) of size 30 took 8 seconds to run. Estimate how long it would take to process a worst-case input of the following sizes: 32, 40, 50, and 60. Express your estimated time in whatever unit makes the most sense (seconds, hours, days, years, ...).