Please hand in each problem on separate sheets with your name and netID on each. If a problem requires multiple sheets, please staple the sheets for that problem together.

Reading: Chapters 6.1 - 6.5.

Question 1

Suppose you work in a novelty beer factory, monitoring the quality of funny shaped beer containers as they roll by on a conveyor belt. You haven’t been so great about getting to work on time (9 am is early!), so you’re pretty sure that you’ll be fired by the end of the day. It’s OK, you didn’t like the job anyway. But you’re still a little indignant and decide to stick it to the man. You figure you’ll steal as much beer as possible before you get kicked out on the street. The only problem is Myrtle. She also monitors the conveyor belt a little ways down. She can’t see so well, but you’re pretty sure she’d notice and sound the alarm if you took all the beer for the day. But maybe if you didn’t take too much at any time, things would be OK.

You decide that to play it safe, you will never take three consecutive beers off the conveyor belt. Now you just need to figure out which ones to take to get the most booze; the thing is, the containers come in all different sizes. Luckily for you, you’ve worked here long enough to know which sizes come when.

OK, so \( n \) beer bottles will appear in order, one at a time, over the course of the day. Each bottle \( i \) has a volume \( v_i \). Give an algorithm that determines the maximum volume of beer that can be stolen without ever taking three consecutive bottles.

Example If \( n = 5 \) with \( v_1 = 3 \), \( v_2 = 8 \), \( v_3 = 9 \), \( v_4 = 4 \), and \( v_5 = 8 \) then the optimal solution is to steal bottles 2, 3 and 5, for a total volume of 25. You could steal more bottles by taking 1, 2, 4 and 5 but this only yields a total volume of 23.

Question 2

Having been fired from the beer company, you start working in a rectangle-making company. That’s right, you can’t even get a job working for a box-making company: Your new employer makes rectangles. Of all shapes and sizes. Well, of all sizes, at least. In an effort to boost sales (which, surprisingly, are lagging), the company would like to emphasize the convenient nesting property of their rectangles. That is, if one rectangle is smaller than another in both width and height, then the smaller rectangle can be nested inside the bigger one. Neat, huh? Your boss would like you to determine the maximum number of rectangles than can be nested one inside another.

To be more concrete, assume the company produces \( n \) different rectangles. Rectangle \( i \) has a width \( w_i \) and a height \( h_i \). Rectangle \( i \) can fit inside rectangle \( j \) if \( w_i < w_j \) and \( h_i < h_j \).

Assume for simplicity that rectangles can not be rotated 90 degrees (so a \( 2 \times 5 \) rectangle does not fit inside a \( 6 \times 3 \) rectangle).

Give an algorithm that selects the largest possible set of nested rectangles. That is, the maximum number of rectangles with the property that the first fits inside the second, the second fits inside the third, etc.
Question 3

You’ve just finished building your time machine, and decide to put it to good use. You travel into the future, grab a book containing the monthly stock prices from Feb. 2006 onward, and then come back to the present. You’d like to use that information to become absurdly wealthy. You’ve found two companies, which we will call company A and company B, whose stock prices have gone up for every month you have on record. To keep things simple, you give yourself the following restrictions. At any given time, your money will be invested in exactly one of these two companies, and every month you’ll choose to either keep your money invested as it is, or switch to the other company.

At first, it seems like this is really simple; you’ll just put your money in the stock that does better in any given month. However, there is a catch. You have to pay a flat commission fee to buy and sell stocks, so there is a cost of switching your investment.

To be formal, you decide you want to invest for $n$ months. You initially have $m$ dollars invested in A. Investing in stock A on month $i$ multiplies your money by $a_i > 1$. Similarly, investing in stock B on month $i$ multiplies your money by $b_i > 1$. To move your money between stocks, however, incures a fixed cost, $c > 0$. We will ignore any tax issues (if you can build a time machine, you can probably figure out how to get around the IRS).

Give an algorithm to determine the maximum amount of money you can accrue by the end of $n$ months.

Example Suppose $n = 2$, $m = 100$ and $c = 20$. The company A has growth $a_1 = 1.1$ and $a_2 = 2$, while company B has growth $b_1 = 1.5$ and $b_2 = 1.8$. You have four options, since you can have your money invested in either company on each month. Let’s consider two possibilities. If you switch your investment from A to B on the first month, and then switch back to A for the second month, your return is $((100 - 20) \times 1.5 - 20) \times 2 = 200$. However, if you just keep your money in A for both months, your return is $100 \times 1.1 \times 2 = 220$. This turns out to be the best you can do for this example.

Hint: Try defining two sets of subproblems, say $OPT_A(i)$ and $OPT_B(i)$, for all $i$ from 1 to $n$. 