CS211 Final

Due Friday, April 10, 5 p.m. (End of Exam Period)

Each problem has multiple parts, and each part involves multiple steps. For all problems, be very clear in your write up and show your work so that I can understand how you arrived at a solution.

1. (20 pts) You are given a pile of coins that all look the same. They are the same size, the same color and the same engravings. You are also told that exactly one of the coins is fake. The fake coin can be distinguished from real coins because it weighs less than real coins. You are also given a balance that you can use to compare the weights of coins. Each side of the balance is large enough to hold the entire pile of coins.

   Design an efficient algorithm to identify the fake coin. Analyze your algorithm in terms of the number of weighings you perform.

2. (20 pts) In class, we discussed a greedy algorithm to produce change, shown below. (In the algorithm, we did not make the output explicit, but we could imagine printing out the change or returning a Change object.) This algorithm does not always produce change using the fewest number of coins for more unusual coin systems.

Algorithm 1 GIVECHANGE(amount)

```
numQuarters = amount / 25
amount = amount % 25
numDimes = amount / 10
amount = amount % 10
numNickels = amount/5
amount = amount % 5
numPennies = amount
```

   (a) Show that the greedy algorithm is not optimal for coins valued at 4, 3, and 1.

   (b) Design an algorithm that will produce the fewest number of coins as change, given coins valued at c_1 to c_k and producing change in any value from 1 to n. You may assume that the smallest-valued coin has a value of 1 so that any amount of change can be produced. You can also assume that you have an unlimited supply of each type of coin.

   (c) Work through the example of producing change worth 10 for a coin system where the coins are valued at 6, 5 and 1 using the algorithm you designed.

   (d) Analyze your algorithm's running time.

3. (20 pts) You are a tournament director and need to arrange a tournament among $N = 2^k$ players. Each player plays exactly one game every day. After $N - 1$ days, each player will have played each other player exactly once.

   (a) Design an efficient algorithm to solve this scheduling problem.

   (b) Show the solution produced by your algorithm for the players Alice, Bob, Carol and David. (You may call them by their nicknames A, B, C, and D if you prefer.)

   (c) Analyze your algorithm's running time.

4. (20 pts) 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. Since you're still a little indignant, 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.

   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 that only yields a total volume of 23.

   (a) In the example above, a greedy algorithm that sorted by bottle weights would arrive at the optimal solution. Show a counterexample where such a greedy algorithm would not provide an optimal solution.

   (b) Design an algorithm to solve the problem and analyze your algorithm.

   (c) Show the results of using your algorithm on the example above.

Extra Credit - 10 pts

It's around the time of the opening day of baseball season, but you're already looking forward to the World Series.

The Yankees have made it to the World Series against your favorite team. The World Series is a best of 7 series which means that the first team to win 4 total games is declared the winner. Thus, the series can be as short as 4 games or as long as 7 games. As an amateur gambler, you plan to place bets on each of the games in the series. Unfortunately, your gambling exploits from the NCAA Basketball tournament have left you with only \$100 in your pocket. While your love for your team is unbounded, so too is your enmity for the Yankees. This acrimony has led you to the following decision If the Yankees win, you want to lose all \$100, but if your team wins, you want to double your money. What should your strategy be? In particular, how much money should you bet on the first game? Note: There is no probability in this question—your strategy is based purely on the wins and loses of the two teams in the series.

1. Start by letting $p(i, j)$ be your current winnings or losings when your team has i wins and the Yankees have j wins. For example $p(4, 1) = 100$ because if your team wins you should win \$100 dollars while $p(1, 4) = -100$ since if your team loses you should lose \$100. I In general, what are the base cases for p?

2. Write a recursive definition for $p(i, j)$.

3. Now, $p(0, 0)$ should be 0, so $p(1, 0)$ should reveal your bet. What is it?