Problem Set 1: Due Monday, February 26, 2018 by 10:10am
(within the first 5 minutes of class)

Homework Instructions
Much of what is learned in this course will be from trying to solve the homework problems, so make a conscientious effort to complete them well.
Your solutions should be **concise** and **clear**. Understandability of the solution is as necessary as correctness. Expect to lose points if you provide a “correct” solution with an unclear write-up. As with an English paper, do not expect to turn in a first draft: it takes refinement to describe something well. If you can’t solve a problem, briefly indicate what you’ve tried and where the difficulty lies. Don’t try to pull one over on us. :)
You may (and are encouraged to) discuss the homework problems with your classmates. For each problem, you must acknowledge the people with whom you discussed your work, and you must independently write up your own solutions (you should not have the same writeup as another student). You must also acknowledge any other sources (online, text, etc.) that you use.

Your homework submission must:
- Be typed, stapled, and submitted as a hardcopy within the first 5 minutes of class on the due date.
- Include the names of any people with whom you discussed your work or any other sources (online, text, etc.) that you used.
- Include the honor code with your signature on the top of the first page (this acknowledges that you are not copying the solutions from other students or online sources).

---

**Problem 1.** Algorithm Run-Times

Suppose you have to solve a problem and have a choice of four algorithms that have the following approximate running times as a function of input size \( n \):

- Algorithm 1 takes \( 2^n \) seconds
- Algorithm 2 takes \( 32n^2 \) seconds
- Algorithm 3 takes \( 128n \log_2 n \) seconds
- Algorithm 4 takes \( 4096n \) seconds

For each algorithm above, give the range of \( n \) for which it will be the best (i.e. most efficient) algorithm to use. You need only consider values of \( n > 1 \). (Note \( \log n \) is \( \log \) base 2 of \( n \)). Please use the following format for your answers:

Algorithm 1 is optimal for . . .
Algorithm 2 is optimal for . . .

(3 points for each algorithm)

---

**Problem 2.** Coin Changing variations

(a) In class, we considered the following program for solving the memoized version of the Coin-Changing problem:

```
www.cs.middlebury.edu/~achristman/cs302/Programs/CoinChangeMemo.java
```

The program returns only the number of total coins used, not the actual change to be made (e.g. for \( n = 85 \) cents it returns 4, not 3 quarters and 1 dime).

Modify the program so that it computes and prints the actual change for the given amount (e.g. as 25, 25, 25, 1), and not just the minimum number of coins. Ideally your modifications should not have a major effect on the running time (but you may use some extra space to store information to return the more detailed answer). Submit your program via the link on canvas.
(b) Assuming that the coins are only quarters, dimes, nickels, and pennies, give pseudocode for a constant-time algorithm that solves the Coin-Changing problem. You may assume (somewhat unrealistically) that you can do standard arithmetic operations on any size numbers in $O(1)$ time.

Complete Problem 4-1(d) (7 points), and (g) (13 points) on page 107 (a copy of the text is on reserve in the Armstrong Library).

[12] Problem 4. Recurrences
Complete Problem 4-3 (f) on page 108.