Eric's Comments on Problem Set 3.

Problem 1:

In Dr. Jiang's hints, he poses the following question: How many terms must be included in the width, which is given by

$$\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n} \text{ yards,}$$

such that the width exceeds the length in an area of one acre (= 4840 yd^2).

This seems like a straightforward problem, but it turns out that the number of terms that must be included in the sum is truly enormous, and the sum is beyond the capabilities of even a modern high-speed computer. The reason is that the sum grows very, very slowly. The sum is given to an excellent approximation by

$$\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n} \approx \ln\left(n+\tfrac{1}{2}\right)+\gamma ,$$

where ln is the natural logarithm function, or log to the base e, and $\gamma = 0.57721566490\ldots$ is Euler's constant. If you have a scientific calculator, you will find the ln function on one of the buttons, and you should also find the inverse function, which looks like e^x. With this approximation you can get an excellent idea of what n you need.

Algorithm 1

This method is a bit confusing, but if you systematize it, it is the most efficient way to calculate the decimal square root of a number by hand. Here is the method laid out for efficient calculation. It resembles long division, with a bit of a twist. The main difference is that what would normally be the divisor is changing (and getting longer) at each step. I will illustrate by finding $\sqrt{1234.567}$.

```
    3 5. 1 3 6 4 0 5 6
3 √1234.567000000000000
     ^  ^  ^ ^ ^ ^ ^ ^
3   9
65  334
 5  325
701   9.56
  1   7.01
7023  2.5570
   3  2.1069
70266   450100
    6   421596
702724   2850400
     4   2810896
70272805   395040000
      05   351364025
702728106   4367597500
        6   4216368636
702728112_   151228864
```

First, write the decimal number you are trying to take the square root of under a radical sign (where the dividend would normally go). Now group the digits by pairs, starting at the decimal point. I've illustrated the resulting groupings with little carats (^) separating the digits. Start with the leading digit or pair of digits (12, in this case). Find the largest integer a whose square is less than the leading digit or digits (3 in this case). Write this both above the leading digit(s), and to the left of the radical sign (where the divisor goes). Now multiply them and subtract, just as you would when performing ordinary long division (12 – 9 = 3). Bring down two new digits (34). Now, copy the digit you just wrote down and put it below the divisor (3). Add this to the divisor, and write this below (3+3=6). Now draw a blank space to the right of this number. This is your new divisor (6_).

In a manner similar to ordinary long division, guess how many times the new divisor (6_) goes into the current remainder (334). Since sixty-something goes into 334 about five times, we guess five. Fill in the number 5 both up where the quotient would go, and in the blank space to the right of the 6, making it 65. Multiply the new digit (5) by the running divisor (65) and subtract the result (325), to give the new remainder (9). Copy down two more digits. (Normally this would appear as 956, without the decimal point, but I've kept the decimal to help make the columns line up.)

The process can be repeated as many times as you want, just like long division. The divisor keeps getting longer and longer. Note that the answer appears where the quotient would normally appear. Note that the digits in the answer are always spaced out twice as far as the other digits, and that the decimal point lines up with the decimal in the original problem. Note also that any time you get a zero digit, you can skip some steps, and just bring down two more digits without doing the multiplication, subtraction, and division, as illustrated in the step where we added the digits 05. In this case, the answer is 35.1364056… .

Algorithm 2 and Algorithm 4

First of all, these are identical algorithms. If you look at the formula for $\tilde{b}$ in terms of a or b, it is given by

$$\tilde{b} = \frac{a}{2} + \frac{x}{2a} = \frac{b}{2} + \frac{x}{2b},$$

which is identical to the equation for a_{n+1} in terms of a_n. Hence if either of these methods work, they will both work.

In both of these methods, it is not necessary to pick a (or a_1) to be the first digit of the square root. Indeed, any approximation will work (and work pretty well). For example, if you are trying to find $\sqrt{3}$, you will do better to start with 2 than with 1.

Newton's Method

The usual argument for Newton's method involves calculus, but it does not need to. Suppose you are trying to find the n'th root of x, and you have a pretty good estimate a of this root; that is, a^n is pretty close to x. It is a reasonable guess that the correct solution is pretty close to a, in other words, it is $a + h$, where h is pretty small. Hence the goal is to solve the equation

$$(a+h)^n = x$$

Now, finding *h exactly* is no easier than the original problem, but we can approximate h as follows. Expand out $(a+h)^n$. Now, in this expansion, since h is supposed to be small, it makes sense to figure that the more factors of h there are in a term, the smaller the term will be. To get Newton's method to work, keep *only* terms that have zero or one h in them. Solving for h is then simple. You than make $a + h$ your new approximation for a, and you repeat the process.

Newton's method is used in lots of real world applications when the functions are too complicated to solve exactly. It is powerful because, once you get a pretty good approximation to the solution, it roughly doubles the number of correct digits at each step.