ECE 298JA **NS #2 – Version 1.28 April 22, 2018** **Fall 2017**

Univ. of Illinois **Due Wednesday, Sept 13, 2017** **Prof. Allen**

Topic of this homework: Prime numbers, greatest common divisors, the continued fraction algorithm **Corrections from orginal posting in blue.**

Deliverable: Answers to questions.

1 Prime numbers

1. According to the fundamental theorem of arithmetic, every integer may be written as a product of primes.
   (a) Put the numbers $1,000,000$, $1,000,004$ and $999,999$ in the form $N = \prod_k \pi_k^{\beta_k}$ (you may use Matlab to find the prime factors).
   (b) Give a generalized formula for the natural logarithm of a number, $\ln(N)$, in terms of its primes π_k and their multiplicities β_k. Express your answer as a *sum* of terms.
2. Explain why the following 2-line Matlab/Octave program returns the prime numbers π_k between 1 and 100?

```
n=2:100;
k = isprime(n);
n(k)
```

3. Prime numbers may be identified using 'sieves'
   (a) By hand, perform the sieve of Eratosthenes for $n = 1 \dots 49$. Circle each prime p then cross out each number which is a multiple of p.
   (b) In part (a), what is the highest number you need to consider before only the primes remain?
   (c) Generalize: for $n = 1 \dots N$, what is the highest number you need to consider before only the primes remain?
   (d) Write each of these numbers as a product of primes:
   22=
   30=
   34=
   43=
   44=
   48=
   49=
   .
4. Find the largest prime $\pi_k \le 100$? Hint: Do not use matlab other than to check your answer. Hint: Write out the numbers starting with 100 and counting backwards: 100, 99, 98, 97, $\cdots$. Cross off the even numbers, leaving $99, 97, 95, \cdots$. Pull out a factor (only 1 is necessary to show that it is not prime).
5. Find the largest prime $\pi_k \le 1000$? Hint: Do not use matlab other than to check your answer.

2 Greatest common divisors

Consider the Euclidean algorithm to find the greatest common divisor (GCD; the largest common prime factor) of two numbers. Note this algorithm may be performed using one of two methods:

Method	Division	Subtraction
On each iteration...	$a_{i+1} = b_i$ $b_{i+1} = a_i - b_i \cdot \text{floor}(a_i/b_i)$	$a_{i+1} = \max(a_i, b_i) - \min(a_i, b_i)$ $b_{i+1} = \min(a_i, b_i)$
Terminates when...	$b = 0$ (gcd$= a$)	$b = 0$ (gcd$= a$)

We recommend the division method (Eq. 2.1, Sect. 2.1.2, Lec 5, Ch. 2), given in the course notes.

1. Understand the Euclidean (GCD) algorithm
   - (a) Use the Matlab command `factor` to find the prime factors of $a = 85$ and $b = 15$. What is the greatest common prime factor of these two numbers?
   - (b) By hand, perform the Euclidean algorithm for $a = 85$ and $b = 15$.
   - (c) By hand, perform the Euclidean algorithm for $a = 75$ and $b = 25$. Is the result a prime number?
   - (d) Consider the first step of the GCD division algorithm when $a < b$ (e.g. $a = 25$ and $b = 75$). What happens to a and b in the first step? Does it matter if you begin the algorithm with $a < b$ vs. $b < a$?
   - (e) Describe in your own words how the GCD algorithm works. Try the algorithm using numbers which have already been separated into factors (e.g. $a = 5 \cdot 3$ and $b = 7 \cdot 3$).
2. Coprimes
   - (a) Define the term *coprime*.
   - (b) How can the Euclidean algorithm be used to identify coprimes?
   - (c) Give at least one application of the Euclidean algorithm.
3. Write a Matlab function, `function x = my_gcd(a,b)`, which uses the Euclidean algorithm to find the GCD of any two inputs `a` and `b`. Test your function on the (a,b) combinations from parts (a) and (b). Include a printout (or handwrite) your algorithm to turn in.

   Hints and advice:

   - Don't give your variables the same names as Matlab functions! Since `gcd` is an existing Matlab/Octave function, if you use it as a variable or function name, you won't be able to use `gcd` to check your `gcd()` function. Try `clear all` to recover from this problem.
   - Try using a 'while' loop for this exercise (see Matlab documentation for help).
   - You may need to make some temporary variables for `a` and `b` in order to perform the algorithm.

3 Alegbraic generalization of the GCD (Euclidean) algorithm

In this problem we are looking for integer solutions $(m,n) \in \mathbb{Z}$ to the equations $\mathbf{ma}+\mathbf{nb} = \mathbf{gcd}(\mathbf{a},\mathbf{b})$ and $\mathbf{ma}+\mathbf{nb} = \mathbf{0}$ given positive integers $(a,b) \in \mathbb{Z}^+$. Note that this requires that either m or n be negative. These solutions may be found using the Euclidean algorithm only if (a,b) are coprime $(a \perp b)$. Note that integer (whole number) polynomial relations such as these are known as 'Diophantine equations.' The above equations are *linear Diophantine equations*, possibly the simplest form of such relations.

Matrix approach: It can be difficult to keep track of the a's and b's when the algorithm has many steps. Here is an alternative way to run the Euclidean algorithm, using matrix algebra. Matrix methods provide a more transparent approach to the operations on (a,b). Thus the Euclidean algorithm can be classified in terms of standard matrix operations (discussed in Lec. 5 pp. 73-75):

Division method:
Define

$$\begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \qquad \begin{bmatrix} a_{i+1} \\ b_{i+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -\lfloor a_i/b_i \rfloor \end{bmatrix} \begin{bmatrix} a_i \\ b_i \end{bmatrix}$$

Example: `gcd(2,3)=1`**):** For $(a,b) = (2,3)$, the result is as follows:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 1 \\ 3 & -2 \end{bmatrix}}_{m \quad n} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Thus from the above equation we find the solution (m,n) to the integer equation

$$2m + 3n = \texttt{gcd}(2,3) = 1,$$

namely $(m,n) = (-1,1)$ (i.e., $-2+3=1$). There is also a second solution $(3,-2)$, (i.e., $3 \cdot 2 - 2 \cdot 3 = 0$), which represents the terminating condition. Thus these two solutions are a pair and the solution only exists if (a,b) are coprime $(a \perp b)$.

Subtraction method:
This method is more complicated than the division algorithm, because at each stage we must check if $a < b$. Define

$$\begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \qquad Q = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \qquad S = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

where Q sets $a_{i+1} = a_i - b_i$ and $b_{i+1} = b_i$ assuming $a_i > b_i$, and S is a 'swap-matrix' which swaps a_i and b_i if $a_i < b_i$. Using these matrices, the algorithm is implemented by assigning

$$\begin{bmatrix} a_{i+1} \\ b_{i+1} \end{bmatrix} = Q \begin{bmatrix} a_i \\ b_i \end{bmatrix} \text{ for } a_i > b_i, \qquad \begin{bmatrix} a_{i+1} \\ b_{i+1} \end{bmatrix} = QS \begin{bmatrix} a_i \\ b_i \end{bmatrix} \text{ for } a_i < b_i.$$

The result of this method is a cascade of Q and S matrices. For $(a,b) = (2,3)$, the result is as follows:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{S} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}}_{m \quad n} \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Thus we find two solutions (m,n) to the integer equation $2m + 3n = \text{gcd}(2,3) = 1$.

1. By inspection, find at least one integer pair (m,n) that satisfies $12m + 15n = 3$.

2. Using matrix methods for the Euclidean algorithm, find integer pairs (m, n) that satisfy $12m + 15n = 3$ and $12m + 15n = 0$. Show your work!!!

3. Does the equation $12m + 15n = 1$ have integer solutions for n and m? Why, or why not?

4 Continued fractions

Here we explore the continued fraction algorithm (CFA), as discussed in Lec. 6 (Chapters 1 and 2). In its simplest form the CFA starts with a real number, which we denote as $\alpha \in \mathbb{R}$. Let us work with an irrational real number, $\pi \in \mathbb{I}$, as an example, because its CFA representation will be infinitely long. We can represent the CFA coefficents α as a vector of integers n_k, $k = 1, 2 \ldots \infty$

$$\begin{aligned} \alpha &= [n_1., n_2, n_3, n_4, \ldots] \\ &= n_1 + \cfrac{1}{n_2 + \frac{1}{n_3 + \frac{1}{n_4 + \ldots}}} \end{aligned}$$

As discussed in Section 1.2.5, the CFA is recursive, with three steps per iteration:

For $\alpha_1 = \pi$, $n_1 = 3$, $r_1 = \pi - 3$ and $\alpha_2 \equiv 1/r_1$.

$$\alpha_2 = 1/0.1416 = 7.0625 \ldots$$
$$\alpha_1 = n_1 + \frac{1}{\alpha_2} = n_1 + \frac{1}{n_2 + \frac{1}{\alpha_3}} = \ldots$$

In Matlab (Octave) script

```
alpha0 = pi;
K=10;
n=zeros(1,K); alpha=zeros(1,K);
alpha(1)=alpha0;

for k=2:K  %k=1 to K
n(k)=round(alpha(k-1));
%n(k)=fix(alpha(k-1));
alpha(k)= 1/(alpha(k-1)-n(k));
%disp([fix(k), round(n(k)), alpha(k)]); pause(1)
end
disp([n; alpha]);
%Now compair this to matlab's rat() function
rat(alpha0,1e-20)
```

1. By hand (you may use Matlab as a calculator), find the first 3 values of n_k for $\alpha = e^{\pi}$.
2. For part (1), what is the error (remainder) when you truncate the continued fraction after $n_1, ..., n_3$? Give the absolute value of the error, and the percentage error relative to the original α.
3. Use the Matlab program provided to find the first 10 values of n_k for $\alpha = e^{\pi}$, and verify your result using the Matlab command `rat()`.
4. Discuss the similarities and differences between the Euclidean algorithm (EA) and CFA.