HW 9 CMSC 456. Morally DUE Nov 23

1. (30 points) Let p, q, r be distinct primes. Let $A = \{1, \ldots, pqr\}$.

   (a) (7 points) How many numbers in A are multiples of p?
   How many numbers in A are multiples of q?
   How many numbers in A are multiples of r?

   (b) (7 points) How many numbers in A are multiples of pq?
   How many numbers in A are multiples of pr?
   How many numbers in A are multiples of qr?

   (c) (1 points) How many numbers in A are multiples of pqr?

   (d) (15 points) Find a formula for $\phi(pqr)$ (use the above answers and the law of inclusion and exclusion.)

2. (30 points) Alice and Bob are trying to do RSA but in a slightly different way. Alice is going to generate THREE primes of length L, which they call p, q, r and then use $N = pqr$. They call this 3-prime-RSA. And they are proud of it!

   (a) (6 points) Fully describe 3-prime-RSA: What Alice does to set it up, what Bob does to send a message, what Alice does to decode a message.

   (b) (6 points) Alice picks $p = 7$, $q = 11$, $r = 13$, and $e = 17$. What is R? What is d? State the actual numbers Alice makes public.

   (c) (6 points) Bob wants to send plaintext 19. What does he send as ciphertext?

   (d) (12 points) Assume Eve knows the Pollard-Rho algorithm, but not the fancier factoring algorithms. Is 3-prime-RSA less secure than the usual RSA? Explain your answer. (Assume that the length of N in both is the same.)

3. (40 points) This is a programming assignment where you will code up the Pollard-Rho algorithm. The same programming languages and file naming conventions from previous homeworks apply.

   (a) Write a program that will factor numbers.

   The first line of stdin will contain an integer n.

   The next n lines will each contain an integer x. For each x, print the prime factors of x as a space-separated list on its own line.

   x may be prime, and have only one factor, or x may have 3 or more factors. For the sake of this assignment, x is at most $2^{127} - 1$. (If you really want to use C/C++, the non-standard __int128 type is available on the grade server.)

   (b) Run some tests. Generate some random primes. Multiply them against each other. Factor the result with your algorithm, and time it. For each of the following cases, collect a bunch of data points (ideally 100 or more):

   - Generate a bunch of primes of diverse orders of magnitude. Multiply pairs of primes, then factor the product and time how long it takes. You should have data points where primes of similar sizes are being multiplied, but also points where one prime is much bigger than the other.
     Plot the time it takes to factorize vs the size of the product, plot time taken vs the smaller prime, plot time taken vs the bigger prime. Fit your graph(s) to to what you know about the runtime of Pollard-Rho.
   - Do the same thing, but multiply 3 primes instead of 2 primes. Use this to conjecture about the runtime for 3 factors.