All exercise numbers from the textbook refer to the **second edition**.

Written problems

1. Use the babystep-giantstep algorithm to compute each of the following discrete logarithms. Show your calculations, e.g. in the form of the table on page 83 of the textbook.

   (a) $\log_{10}[13]_{17}$ (that is, solve $10^x \equiv 13 \pmod{17}$)

   (b) $\log_{15}[16]_{37}$

   (c) $\log_5[72]_{97}$

2. Solve each system of congruences. Your answer should take the form of a single congruence of the form $x \equiv c \pmod{m}$ describing all solutions to the system.

   (a) $x \equiv 1 \pmod{3}$
   $x \equiv 2 \pmod{5}$

   (b) $x \equiv 6 \pmod{11}$
   $x \equiv 2 \pmod{10}$

   (c) $x \equiv 2 \pmod{3}$
   $x \equiv 1 \pmod{10}$
   $x \equiv 3 \pmod{7}$

   (d) $x \equiv 6 \pmod{8}$
   $x \equiv 3 \pmod{9}$
   $x \equiv 17 \pmod{17}$

3. Textbook exercise 2.21 (this provides an alternative proof of the "uniqueness" part of the Chinese remainder theorem from the counting argument I presented in class).

4. Let G be a group, and g an element of order L in G. I will write g^n to be the nth power of g with respect to the group operation.

   (a) Prove that if n is an integer dividing L, then $\mathrm{ord}(g^n) = L/n$.

   (b) Prove that n is an integer relatively prime to L, then $\mathrm{ord}(g^n) = L$.

   (c) Prove that if n is any integer, then $\mathrm{ord}(g^n) = L/\gcd(n, L)$. (Observe that this formula mutually generalizes (a) and (b).)

5. I mentioned in class that in the Elgamal cryptosystem, Bob should create a new (and truly random) ephemeral key (denoted k on page 72 of the textbook) each time he enciphers a message to Alice using her public key. Otherwise, he exposes himself to a "known-plaintext attack" from Eve. In this problem, we will see why he also should not just perform "small variations" on a previously-used ephemeral key.

   (a) Suppose that Bob previously used an ephemeral key k_1 to send Alice a message m_1, and that Eve knows what this message is (as well as the enciphered version that Bob sent to Alice). Suppose Bob now enciphers another message m_2 to Alice, using the ephemeral key $k_2 = k_1 + 1$. Explain how Eve can efficiently detect that this is how Bob has obtained k_2 from k_1 (without necessarily determining what k_1 is), and efficiently extract the message m_2.

   (b) Suppose instead that Bob obtains k_2 as $42k_1$. Explain how Eve can efficiently detect this, and extract the message m_2.

   (c) Suppose instead that Bob obtains k_2 as $k_1 - 5$. Explain how Eve can efficiently detect this, and extract the message m_2.

(d) Suppose instead that Bob obtains k_2 as $13k_1 + 2$. Explain how Eve can efficiently detect this, and extract the message m_2.

Note. This problem seems make what looks like a strange assumption: that Eve knows the plaintext of a previously-sent message to Alice. In fact, this happens quite often; for example, Bob might sent the same message to many people (e.g. a boilerplate introduction or header information), including both Eve and Alice. For this reason, it is important to make sure that cryptosystems used in practice are not vulnerable to these so-called "known-plaintext" attacks.

6. Write a function to do the following task: generate a sequence of random B bit nonnegative numbers (that is, integers with $0 \leq n < 2^B$, where B is given as input), until a number is repeated. The function should return the number of numbers generated (including the repeat at the end). Write a second function that runs your first function 1000 times and averages the results. Run your second function for all values of B from 1 to 20 and report the results. See if you can identify a pattern. (You may already know what pattern to expect to see; the result is slightly counterintuitive and is referred to as the "birthday paradox." This pattern turns out to be of crucial importance in studying the danger posed by certain randomized attacks on cryptosystems.)

7. Go to the following demonstration assignment, and open the problem called "DLP benchmarking." This problem allows you to benchmark the effectiveness of different algorithms to solve the discrete logarithm problem.

   `https://www.hackerrank.com/m158-2016-demos/`

   Note that case number n uses a prime of length exactly $\lfloor \frac{n}{2} \rfloor + 17$ bits. So by seeing which cases a particular algorithm solves, you can gauge the length of prime it is able to handle within the hackerrank time limit.

   (a) Before submitting any code, estimate the number of test cases that you think a naive trial-and-error approach (i.e. testing all possible exponents, starting from 0, until one is found that works) will correctly solve. Then implement such an approach (perhaps using your submission to Problem Set 3, if you used a trial-and-error approach there), submit it, and check how close your estimate was.

   (b) Estimate how many cases an implementation of Babystep-Giantstep will complete correctly. If you have a working BSGS implementation (e.g. after completing the coding portion of the assignment), use it to check your answer, but do not submit the code until you have made your estimate.

   Note. For this assignment, you will receive full point if you make a good faith effort and your reasoning makes sense. I may ask estimations like this on future exams; in this case, I would mark your answer correct if you estimate the number of bits (in the length of p) that the program can handle within 10 bits of the actual figure.

Programming problems

Full formulation and submission: `https://www.hackerrank.com/m158-2016-pset-4`

8. Solve the discrete logarithm problem, where the modulus is a 36 bit prime number.

9. From a list of congruences $x \equiv a_i \pmod{m_i}$, where the integers m_i are pairwise relatively prime, determine integers a, m such that the list is equivalent to the single congruence $x \equiv a \pmod{m}$.

10. Alice and Bob use Diffie-Hellman key exchange on a regular basis, but they are not choosing their secret numbers a and b randomly. As a result, the secret numbers that Alice chooses during different key exchanges are usually close to each other; Bob make the same mistake. More precisely: you may assume that if Alice uses a_0 as her secret number one day and a_1 on another day, then $|a_0 - a_1| \leq 2^{20}$ (and similarly with Bob's numbers).

    Eve has managed to learn one of Alice and Bob's previous shared secret S_0, corresponding two exchanged numbers A_0, B_0 (from Alice to Bob and vice versa). Later, she intercepts two more exchanged numbers A_1, B_1, and wishes to extract the new shared secret S_1 corresponding to these. From all of this information, and the knowledge about how Alice and Bob are choosing their secret numbers, determine S_1.

    Hint. Use similar ideas to those used in problem 5.