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

1. (a) Textbook exercise 3.3 (this shows, as we mentioned in class, that RSA decryption always works when the modulus is a product of two primes, regardless of whether the original message was a unit modulo N or not).

   (b) Suppose that Bob publishes a public key $N = 117, e = 5$ (note that this is not an RSA public key, since 117 isn't a product of two primes), and he requests that Alice encrypt her message m (where $0 \leq m < N$) by sending him $c \equiv m^e \pmod{N}$, as in RSA. Find two different messages m_1, m_2 which give the *same* ciphertext. This shows that Bob *cannot* always decrypt messages unambiguously in this system. (Try to find m_1 and m_2 using a principle that you would be able to generalize, rather than by guess-and-check).

   Note. In general, unique decryption will be possible as long as N is *squarefree*, meaning it is not divisible by any square. For example, decryption is always possible if $N = pqr$, where p, q, r are distinct prime (see problem 11).

2. Textbook exercise 3.6.

3. Textbook exercise 3.10.

4. Textbook exercise 3.17.

5. Textbook exercise 3.18.

6. For each integer n between 1,000,000 and 1,000,009 inclusive, determine the proportion of the numbers from 1 to $n - 1$ inclusive that are Miller-Rabin witnesses. Which of these numbers are prime? (The figures you obtain should convince you that the 75% figure from Rabin's theorem is rather conservative, and explains why most people are not worried about using only a few Miller-Rabin trials to test primality).

7. Textbook exercise 3.22. (We will discuss Pollard's algorithm in class on Monday).

8. Suppose that p is a large prime (e.g. 1024 bits), g is a primitive root modulo p, and Alice has an Elgamal public key A corresponding to a private key a (that is, $A \equiv g^a \pmod{p}$, and Alice knows the number a). Bob does not believe that Alice actually knows the private key a corresponding to A, so she asks her to solve the following challenge to prove it. Bob will give Alice a positive integer d of his choosing. Alice must return (in a reasonable amount of time) *two* integers b, c such that

   $$g^b \cdot b^c \equiv A^d \pmod{p}.$$

   If she succeeds, Bob will be convinved that Alice really does know her private key.

   (a) Describe a procedure that Alice can use to solve Bob's challenge efficiently.
   Hint. Choose an integer e at random, and choose b to be $g^e \pmod{p}$. Then find a choice of c.

   (b) Explain briefly why Bob should be convinced that Eve (or anyone else who doesn't know the private key) would not be able to carry out the procedure you describe in part (a).

   Note. This exercise prefigures the basic idea behind Elgamal "digital signatures," which we will discuss soon. You can solve this problem without knowing anything about signatures, however.

Programming problems

Full formulation and submission: `https://www.hackerrank.com/m158-2016-pset-6`

9. Determine whether a given integer N (up to 1024 bits long) is prime or not.

10. Generate two prime numbers p, q, of specified length (number of bits), such that $p \equiv 1 \pmod{q}$, along with a number $g \in \{1, 2, \cdots, p-1\}$ such that $\text{ord}[g]_p = q$.

    Hint. See exercise 1.33 in the textbook for a useful fact that can help you create g (Problem Set 2, number 6).

    Note. Generating p, q, g of this form is important for choosing parameters for the DSA system, which we will discuss soon.

11. Alice decides that she wants to receive messages using a non-standard variant of RSA. Like in the usual RSA, she will choose a public key N, e, where N is a number whose factorization she knows, and $\gcd(e, \phi(N)) = 1$. In this case, she will take $N = pqr$, where p, q, r are distinct primes. To encrypt a message m for Alice ($0 \leq m < N$), Bob computes $c \equiv m^e \pmod{N}$. Given the three primes p, q, r, the number e, and the ciphertext c sent by Bob, recover the original plaintext m.

    Note. While this setup is perfectly functional, in practice it is more efficient to use products of two primes, hence that is the standard. I encourage you to think about why it is more efficient to use only two primes.

12. You will be given an RSA modulus (a product of two distinct primes), 112 bits in length. Factor it if possible. The primes will have been chosen completely at random, i.e. no effort has been made to avoid "weak" primes. As a result, some of the moduli will be susceptible to Pollard's algorithm (which we discuss in class on Monday), but not all. You are *not* expected to solve all, or even most, of the test cases; a score of 10/50 (as listed on hackerrank; this corresponds to 20 out of 100 test cases correct) will be regarded as full points in the actual grading spreadsheet. If you devise a problem to solve more testcases than these, you will receive extra credit.