This assignment will not be due until Friday, May 3, so that you'll have the entire last full week of class to work on it.

Written problems

1. Textbook exercise 7.1 (6.1 in 1st edition) ("1TRU" examples)

   NOTE: the 1st edition has a couple typos in their summary table for this system; see the note after the first programming problem.

2. Suppose that the "1TRU" system (the system from the first section of §6 in the first edtion and §7.1 of the second edition) is modified as follows.

   - Four parameters F, G, M, R are chosen at the beginning, in addition to the modulus q.
   - When Alice makes her private and public keys, she chooses f, g so that $0 \le f < F$ and $M \le g < G$. She then computes her public key h as before.
   - When Bob sends a message, he must choose his message m so that $0 \le m < M$, and he must choose his random number r so that $0 \le r < R$. Then he computes e as before.

   (a) What are the values of F, G, M, R (in terms of q) used in the original version of "1TRU?"

   (b) Suppose Alice computes b from e as in "1TRU." Write an inequality in terms of F, G, M, R, and q that will guarantee that the number b she computes is definitely equal to m.

   (c) Explain why decryption could fail if the requirement $M \le g$ were dropped in Alice's procedure for generating her public key.

   The remaining two problems concern details of the NTRU system, which we won't discuss until Friday's class.

3. Suppose that Alice and Bob are using NTRU with $N = 251$, $q = 131$, $p = 3$, and $d = 6$. How many bits are needed to represent the plaintext? How many bits are needed to represent the ciphertext? What is the message expansion ratio (ratio of ciphertext bits to plaintext bits)?

4. In some implementations of NTRU, rather than fixing one public parameter d, one chooses three different parameters d_1, d_2, d_3, and stipulates that Alice chooses $\mathbf{f}$ from $\mathcal{T}(d_1+1, d_1)$ and $\mathbf{g}$ from $\mathcal{T}(d_2, d_2)$, and Bob chooses $\mathbf{r}$ from $\mathcal{T}(d_3, d_3)$. Assuming that $d_1 \ge d_2 \ge d_3$, determine an inequality of the form $q > \cdots$ to replace the inequality $q > (6d+1)p$ in table 7.3, serving the same purpose in this more general formulation (your inequality should specialize to $q > (6d+1)p$ in the case $d_1 = d_2 = d_3 = d$).

Programming problems

1. You will be given the public parameter q, Alice's public key h, and a ciphertext e for the congruential cryptosystem (which I called "1TRU" in class). Write a function `break1TRU(q, h, e)` that extracts the plaintext (without knowing the private key).

   Half of the test cases will be small enough that a brute force approach should work. For the remaining test cases I suggest that you use Gauss's lattice basis reduction algorithm.

   Note: this system's summary table is p. 351 in the 1st edition, p. 375 in the 2nd edition. The 1st edition's table has two typos: under "Encryption", the first line should say "$m < \sqrt{q/4}$" (not $\sqrt{q/2}$). Under "Decryption," the first line should say "$a \equiv fe \pmod{q}$" (not $\pmod{e}$).

2. In this problem, you will write several functions to work with "convolution polynomials," i.e. the set we have been denoting in class by R. These functions will be needed to implement NTRU decryption. All five functions take an argument N, the number of coefficients in each polynomial. For all five functions, elements of R will be represented in Python as a list of N integers; `f[i]` is the X^i coefficient of the polynomial. You will write all five functions in the same file; the test bank will test all of them.

   The five functions are:

   - `addR(f, g, N)`, which should return $f(x) + g(x)$ (as a list of N integers).
   - `scaleR(c, f, N)`, which takes an integer c, polynomial $f(x) \in R$, and returns the product $c \cdot f(x)$ (scale all coefficients by c).
   - `convolveR(f, g, N)`, which takes two polynomials $f(x), g(x) \in R$, and returns $f \star g$.
   - `reduceR(f, p, N)`, which takes $f(x) \in R$ and returns the result of reducing f modulo p.
   - `centerliftR(f, p, N)`, which takes $f(x) \in R$ and returns the result of "centerlifting" f modulo p (we discuss this concept on Friday).

3. Write a function `decipherNTRU(N, p, q, d, h, f, g, Fp, Fq, e)` that deciphers an NTRU ciphertext, given all of Alice's private information. See the NTRU summary table (p. 394 in 1st edition; p. 419 in 2nd edition) for the notation (unfortunately, there's a lot of it!). In input, all elements of R will be "centerlifted," so they may have both positive and negative coefficients. You should return the plaintext `m` centerlifted modulo p.

   Note that you may discover that you do not need all ten of these arguments in order to compute the result. I have included all of them for completeness. It's also worth pointing out that in principle I don't need to tell you $\mathbf{F}_p$ and $\mathbf{F}_q$, since they can both be computed from $\mathbf{f}$, but I am not expecting you to implement the algorithms needed to do this.