Mathematical Software - Homework 3

Deadline: Sunday, May 9th

For this exercise you should have received this text in .ipynb format. Complete the exercises by modifying this file, and submit the modified version

Exercise 1

Use SageMath to solve the following problems:

(a) Find the roots of the following polynomial over $\mathbb{Q}$:

$$p = 4x^7 + 4x^6 + 3x^5 - 13x^4 - 13x^3 - 9x^2 + 3x + 3 \in \mathbb{Q}[x]$$

(b) Find the roots of the same polynomial p over $\mathbb{R}$ and over $\mathbb{C}$.

(c) Find the determinant, the trace and the characteristic polynomial of the following matrix:

$$A = \begin{pmatrix} -1 & 1 & -1 & 0 \\ 1 & \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

(d) Find a solution to the linear system $A\mathbf{x} = \mathbf{v}$, where A is the matrix above and $\mathbf{v} = (1, 2, 3, 4)$.

Write your code in the cell below.

[ ]:

Exercise 2

After exchanging messages with the RSA protocol seen in class, Alice and Bob decide to meet and play their favorite game: flip a coin. They like this game very much because it does not take long to set it up and they have exactly the same chances of winning.

Unfortunately, due to the COVID-19 pandemic they cannot meet in person, and despite being good friends they don't trust each other enough to play this game via Webex call. Luckily, Alice is an expert in cryptography and she knows how to play this game using the Chinese remainder theorem.

The game plays out as follows:

(A1) Alice picks two large prime numbers p and q, she computes $n = pq$ and sends n to Bob, keeping p and q secret.

(B1) Bob picks a random number a with $1 < a < n$ and $\gcd(a, n) = 1$, computes $b = a^2 \mod n$ and sends b to Alice, keeping a secret.

(A2) Alice computes two numbers x and y such that $x^2 \equiv b \pmod{p}$ and $y^2 \equiv b \pmod{q}$ and she uses the Chinese remainder theorem to compute a number z such that $z \equiv x \pmod{p}$ and $z \equiv y \pmod{q}$, so that $z^2 \equiv b \pmod{n}$. Then she sends z to Bob.

Since n is the product of two primes, there are 4 possible square roots of b modulo n, corresponding to the solutions of the four systems of congruences (one for each possible combination of $\pm$)

$$\begin{cases} z \equiv \pm x \pmod{p} \\ z \equiv \pm y \pmod{q} \end{cases}$$

One of those solutions is a and another is $-a$, and Bob knows them. Alice is picking one of the 4 possible roots at random (she chooses between x and $-x$ and between y and $-y$), so she has 50% chance of picking one that Bob already knows. This corresponds to Alice flipping a coin, and she wins if she picks $\pm a$:

(B2) If $z \equiv \pm a \pmod{n}$, Bob declares to have lost. Otherwise, Bob claims to have won, and as proof he produces one prime factor of n by computing $g = \gcd(n, a + z)$. *(One can prove that in this situation g is always one of the two prime factors of n.)*

Since factoring a number without extra information is very hard, Alice will be convinced that she must have given Bob one of the square roots that he did not know, so she admits the loss.

Now to the actual exercise:

(a) Write the code for the functions A1, B1 and B2 as indicated in the cell below. The function A2 is already written.

(b) Modify the functions B1, A2 and B2 to check that the opponent is not cheating. More precisely:

- In B1, Bob should check that n is not a prime power. *(This is the only way Alice can try to cheat: if she sends Bob a number n that is the product of more than two primes, than she has less than* 50% *chance of winning!)*
- In A2, Alice should check that b is a square modulo n.
- In B2, Bob should check that $z^2 \equiv a^2 \pmod{n}$.

In case cheating is detected, a message should be printed saying that the person is cheating.

[1]:
```
# Alice needs this to compute the square roots
from sage.rings.finite_rings.integer_mod import square_root_mod_prime

def A1():
    # This function must return two distinct primes and their product.

def B1(n):
    # This function must return a random integer a
    # with 1<a<n and gcd(a,n)=1.

def A2(b, p, q):
    x = ZZ(square_root_mod_prime(Integers(p)(b), p))
    y = ZZ(square_root_mod_prime(Integers(q)(b), q))
    return crt(x, y, p, q)

def B2(a, z, n):
    # This function must print out one of two messages:
    # "Bob has lost" if z is congruent to a or -a modulo n.
    # "Bob has won, proof: " followed by a prime factor of n otherwise.
    # In this case the prime must be calculated as explained above.


# This is how the game plays out:
p, q, n = A1()
print("Alice picked n =", n)
print("[[ Alice's secret:", p, q, "]]")
a = B1(n)
b = a^2 % n
print("Bob picked b =", b)
print("[[ Bob's secret:", a, "]]")
z = A2(b, p, q)
print("Alice picked z =", z)
B2(a, z, n)
```

Grading

This homework assignment is worth 20% of your final grade. Exercise 1 is worth 4 points (one for each part) and Exercise 2 is worth 12 points (8 points for part (a) and 4 points for part (b)), for a total of **16 points**.