EXERCISES FOR TUTORIAL 2

These exercises were already suggested in Tutorial 1, we will discuss the answers during Tutorial 2.

Exercises

- 2.2 and 2.3 with SageMath
- 2.9 is the general addition law with
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$
it can be done with SageMath.

Moreover we will do:

- 2.4 with SageMath
- 2.5
- 2.6
- 2.8 with the algorithm of the book (least significant bits first) then with the variant in the slides of Lecture 2 (most-significant bits first)
- 2.7 with SageMath
- using the file `curve_arithmetic.py`, write the double-and-add algorithm of the book page 18 (integer-times-a-point), and the Most-Significant-Bits First version of the lecture (slides of Lecture 02).
  - With the short Weierstrass equation and the affine formulas, check with what would answer SageMath
  - With the Edwards form and the affine Edwards formula: this is too long, it is only optional.
- Points of order 2: Given an elliptic curve
$$y^2 = x^3 + a_2x^2 + a_4x + a_6$$
over a field $\mathbb{K}$, what are the points of order 2 in terms of their coordinates (x, y)? Hint: it means that the tangent at P of order 2 is a vertical. What do you deduce about the factorisation of $f(x) = x^3 + a_2x^2 + a_4x + a_6$ and the coordinates (x, y)? If $x_0 = 0$ is a root of $f(x)$, which conditions on the coefficients a_i ensure to have as many points of order 2 as possible?
- Points of order 3: these are inflexion points. What are the conditions on the a_i to get points of order 3, and what are their coordinates (x, y) in terms of roots of an appropriate polynomial?

Solutions

To know wich imports are needed when writing a `.py` file, the following instructions are useful in the SageMath interactive shell:

```
import_statements(QQ)
import_statements(EllipticCurve)
import_statements(IntegerModRing)
```

The answers are:

```
# ** Warning **: several names for that object: Q, QQ
from sage.rings.rational_field import Q
from sage.schemes.elliptic_curves.constructor import EllipticCurve
# ** Warning **: several names for that object: IntegerModRing, Integers, Zmod
from sage.rings.finite_rings.integer_mod_ring import IntegerModRing
```

Exercise 1. Not done in Spring 2022, given here for completeness.

(a) Show that the constant term of a monic cubic polynomial is the negative of the product of the roots.

Date: Week 2.

(b) Use 1 to derive the formula for the sum of two distinct points P_1, P_2 in the case that the x-coordinates x_1 and x_2 are non-zero, as in Section 2.2 of Washington's book. Note that when one of these coordinates is zero, you need to divide by zero to obtain the usual formula.

Solution 1.

(a) With SageMath:

```
QQa.<a0,a1,a2> = QQ[] # a polynomial ring in 3 parameters
QQx.<x> = QQa[] # a univariate polynomial ring on top of the 3-parameter ring
f = (x-a0)*(x-a1)*(x-a2)
print("f  = {}".format(f))
print("f.coefficients() = {}".format(f.coefficients()))
```

The answer is

```
f  = x^3 + (-a0 - a1 - a2)*x^2 + (a0*a1 + a0*a2 + a1*a2)*x - a0*a1*a2
f.coefficients() = [-a0*a1*a2, a0*a1 + a0*a2 + a1*a2, -a0 - a1 - a2, 1]
```

We obtain that the constant coefficient of $f(x)$ is indeed the negative of the product of the three roots, namely $a_0a_1a_2$.

(b) Denote the two points $P_1(x_1, y_1), P_2(x_2, y_2)$. The slope of the line through them is $m = (y_2 - y_1)/(x_2 - x_1)$. The intersection of the line $L\colon y = m(x - x_1) + y_1$ with the curve equation $E\colon y^2 = x^3 + ax + b$ is

```
QQa.<x1,x2,y1,y2,a,b,l> = QQ[] # a polynomial ring with parameters
QQx.<x> = QQa[] # a univariate polynomial ring on top of the parameter ring
# the curve equation, substituting y by L
m = (y2-y1)/(x2-x1)
Ly = m*(x-x1) + y1
assert Ly(x=x1) == y1
assert Ly(x=x2) == y2
E = -Ly^2 + x^3 + a*x + b
print("E = {}".format(E))
# let's simplify the constant term:
E.constant_coefficient().denominator().factor()
E.constant_coefficient().numerator().factor()
# one obtains
E.constant_coefficient() == -(x2*y1 - x1*y2)^2/(x1-x2)^2 + b
# one can also do the same with m as a parameter l
ly = l*(x-x1) + y1
El = -ly^2 + x^3 + a*x + b
print("E = {}".format(El))
# one obtains
E = x^3 - l^2*x^2 + (2*x1*l^2 - 2*y1*l + a)*x - x1^2*l^2 + 2*x1*y1*l - y1^2 + b
El.constant_coefficient() == -(x1*l - y1)^2 + b
```

We know that the roots of E are x_1, x_2, and x_3 that we are looking for, and that the negative of product of these roots is the constant coefficient of E. Hence,

```
x3 = -E.constant_coefficient()/(x1*x2)
x3 == (x2*y1 - x1*y2)^2/((x1-x2)^2*x1*x2) - b/(x1*x2)
# with the parameter l
x3l = -El.constant_coefficient()/(x1*x2)
x3l == (x1*l - y1)^2/(x1*x2) - b/(x1*x2)
# to be checked with the formula l^2 - x1 - x2
```

To go back to the formula without b, first we replace b in terms of a:

$$b = y_1^2 - x_1^3 - ax_1$$

Then we replace a with the expression

$$y_1^2 = x_1^3 + ax_1 + b \tag{1}$$

$$y_2^2 = x_2^2 + ax_2 + b \tag{2}$$

$$
\begin{aligned}
(1)-(2) \quad &: \quad y_1^2 - y_2^2 = x_1^3 - x_2^3 + a(x_1 - x_2) \\
&: \quad (y_1 - y_2)(y_1 + y_2) = (x_1 - x_2)(x_1^2 + x_1x_2 + x_2^2) + a(x_1 - x_2) \\
&: \quad (y_1 - y_2)(y_1 + y_2)/(x_1 - x_2) = (x_1^2 + x_1x_2 + x_2^2) + a \\
&: \quad a = m(y_1 + y_2) - (x_1^2 + x_1x_2 + x_2^2)
\end{aligned}
$$

$$
\begin{aligned}
b &= y_1^2 - x_1^3 - ax_1 \\
b &= y_1^2 - x_1^3 - x_1(m(y_1 + y_2) - (x_1^2 + x_1x_2 + x_2^2)) \\
&= y_1^2 - x_1(m(y_1 + y_2) - (x_1x_2 + x_2^2)) \\
&= y_1^2 - x_1m(y_1 + y_2) + x_1x_2(x_1 + x_2)
\end{aligned}
$$

and dividing by x_1x_2, one gets

$$b/(x_1x_2) = y_1^2/(x_1x_2) - m(y_1 + y_2)/x_2 + (x_1 + x_2) \ .$$

Now we replace b in x_3:

$$
\begin{aligned}
x_3 &= (x_1m - y_1)^2/(x_1x_2) - b/(x_1x_2) \\
x_3 &= (x_1m - y_1)^2/(x_1x_2) - y_1^2/(x_1x_2) + m(y_1 + y_2)/x_2 - (x_1 + x_2) \\
x_3 &= (x_1^2m^2 - 2x_1y_1m)/(x_1x_2) + mx_1(y_1 + y_2)/(x_1x_2) - (x_1 + x_2) \\
x_3 &= (x_1^2m^2 - 2x_1y_1m + mx_1(y_1 + y_2))/(x_1x_2) - (x_1 + x_2) \\
x_3 &= (x_1^2m^2 - x_1y_1m + mx_1y_2))/(x_1x_2) - (x_1 + x_2) \\
x_3 &= x_1m(x_1m - y_1 + y_2))/(x_1x_2) - (x_1 + x_2) \\
x_3 &= m(x_1m - y_1 + y_2))/x_2 - (x_1 + x_2) \text{ where } y_2 - y_1 = m(x_2 - x_1), \\
x_3 &= m(x_1m + m(x_2 - x_1))/x_2 - (x_1 + x_2) \\
x_3 &= m(mx_2)/x_2 - (x_1 + x_2) \\
x_3 &= m^2 - (x_1 + x_2)
\end{aligned}
$$

as expected. Then we can check the formula for x_3 with SageMath:

```
x3 = x3(b = y1^2 - x1^3 -a*x1)
x3l = x3l(b = y1^2 - x1^3 -a*x1)
x3 = x3(a = m*(y1+y2) - (x1^2+x1*x2+x2^2))
x3l = x3l(a = l*(y1+y2) - (x1^2+x1*x2+x2^2))

x3 == m^2 - x1 - x2
x3l == (x1*l^2 + l*(y2-y1))/x2 -x1 -x2 # it will not simplify with abstract l
```

Exercise 2. The point $(3, 5)$ lies on the elliptic curve $E\colon y^2 = x^3 - 2$, defined over $\mathbb{Q}$. Find a point (not infinity P_∞) with rational, nonintegral coordinates in $\mathbb{Q}$.

Solution 2. Doubling the point $(3, 5)$ will give another point distinct from the point at infinity P_∞.

```
E = EllipticCurve(QQ, [0, -2])
P = E((3,5))
Q = 2*P
R = 3*P
print("E is {}".format(E))
print("P = {} 2*P = {} 3*P = {}".format(P, Q, R))
```

Answers from SageMath are

```
E is Elliptic Curve defined by y^2 = x^3 - 2 over Rational Field
P = (3 : 5 : 1) 2*P = (129/100 : -383/1000 : 1) 3*P = (164323/29241 : -66234835/5000211 : 1)
```

Exercise 3. The points $P = (2, 9)$, $Q = (3, 10)$, and $R = (-4, -3)$ lie in the elliptic curve $E\colon y^2 = x^3 + 73$, defined over $\mathbb{Q}$.

(a) Compute $P + Q$ and $(P + Q) + R$.

(b) Compute $Q+R$ and $P+(Q+R)$. Your answer for $P+(Q+R)$ should agree with the result of part 1. However, note that one computation used the doubling formula while the other did not use it.

Solution 3. For that exercise, one reuses the functions provided in `curve_arithmetic.py`, with the `import` instruction.

```
from curve_arithmetic import *
E = EllipticCurve(QQ, [0, 73]) # the curve will be over QQ
P = E((2,9))
Q = E((3,10))
R = E((-4,-3))

PQ = group_law_affine(P, Q)
PQR = group_law_affine(PQ, R)
print("(a)")
print("P+Q = {}".format(PQ))
print("(P+Q)+R = {}".format(PQR))
QR = group_law_affine(Q, R)
QRP = group_law_affine(QR, P)
print("(b)")
print("Q+R = {}".format(QR))
print("P+(Q+R) = {}".format(QRP))
print("P+(Q+R) == (P+Q)+R: {}".format(QRP == PQR))

print("Checking the answers of Exercise 2.3 with SageMath")

PQ = P+Q
PQR = PQ + R
print("(a)")
print("P+Q = {}".format(PQ))
print("(P+Q)+R = {}".format(PQR))

QR = Q+R
QRP = P + QR
print("(b)")
print("Q+R = {}".format(QR))
print("P+(Q+R) = {}".format(QRP))
print("P+(Q+R) == (P+Q)+R: {}".format(QRP == PQR))

assert QRP == PQR
```

Exercise 4. Let E be the elliptic curve $y^2 = x^3 - 34x + 37$ defined over $\mathbb{Q}$. Let $P = (1,2)$ and $Q = (6,7)$.

(a) Compute $P+Q$.

(b) Note that $P = Q \mod 5$. Compute $2P$ on $E \bmod 5$. Show that the answer is the same as $(P+Q) \bmod 5$. Observe that since $P = Q \bmod 5$, the formula for adding the points mod5 is not the reduction of the formula for adding $P+Q$. However, the answers are the same. This shows that the fact that reduction modulo a prime is a homomorphism is subtle, and this is the reason for the compicated formulas in Section 2.11.

Solution 4.

```
E = EllipticCurve(QQ, [-34, 37])
P = E((1,2))
Q = E((6,7))
PQ = P+Q
print("(a) P+Q = {}".format(PQ))
```

```
print("xP = xQ mod 5: {}, yP = yQ mod 5: {}".format(P[0] % 5 == Q[0] % 5, P[1] % 5 == Q[1] % 5))

P2 = 2*P
print("2P = {} = ({}, {}) mod 5".format(P2, P2[0] % 5, P2[1] % 5))
print("P+Q= {} = ({}, {}) mod 5".format(PQ, PQ[0] % 5, PQ[1] % 5))

print("Now we define the curve over ZZ/5ZZ")
Z5 = IntegerModRing(5)
E5 = EllipticCurve(Z5, [-34, 37])

P = E5((1,2))
Q = E5((6,7))
PQ = P+Q
print("(a) P+Q = {}".format(PQ))

print("xP = xQ mod 5: {}, yP = yQ mod 5: {}".format(P[0] % 5 == Q[0] % 5, P[1] % 5 == Q[1] % 5))

P2 = 2*P
print("2P = {} = ({}, {}) mod 5".format(P2, P2[0] % 5, P2[1] % 5))
print("P+Q= {} = ({}, {}) mod 5".format(PQ, PQ[0] % 5, PQ[1] % 5))
```

and SageMath answer is:

```
(a) P+Q = (-6 : 5 : 1)
xP = xQ mod 5: True, yP = yQ mod 5: True
2P = (929/16 : 28175/64 : 1) = (4, 0) mod 5
P+Q= (-6 : 5 : 1) = (4, 0) mod 5
Now we define the curve over ZZ/5ZZ
(a) P+Q = (4 : 0 : 1)
xP = xQ mod 5: True, yP = yQ mod 5: True
2P = (4 : 0 : 1) = (4, 0) mod 5
P+Q= (4 : 0 : 1) = (4, 0) mod 5
```

Exercise 5. Let (x, y) be a point in the elliptic curve E given by $y^2 = x^3 + Ax + B$. Show that if $y = 0$ then $3x^2 + A \neq 0$. (Hint: What is the condition for a polynomial to have x as a multiple root?)

Solution 5. E is an elliptic curve, that is, the curve is smooth, and in particular, $f(x) = x^3 + Ax + B$ has no multiple roots, in other terms, the derivative shares no root with f. Let's compute the derivative of f: $f'(x) = 3x^2 + A$, and because the derivative is coprime to the cubic polynomial, $f'(x)$ is non-zero for a point $(x, y = 0)$ on the curve. Hence, $3x^2 + A$ is non-zero.

Exercise 6. Show that three (distinct) points on an elliptic curve add to infinity P_∞ if and only if they are colinear.

Solution 6. For that one, Bezout's theorem is required.

Let P, Q and R be three distinct points on a curve E. First, assume that P, Q and R add to infinity. It means, $(P + Q) + R = P_\infty$. Consider the line through P and Q, it intersects the curve to a third point S. Consider the vertical at S, it intersects the curve at another point T. Now add $T + R$: by assumption we get a vertical (so that $T + R = P_\infty$). But there is already S that intersects the curve with the vertical at T. Because a line as degree 1 and the curve has degree 3, the intersection of the vertical with the curve has three intersection points, counting the point at infinity, that is, there are at most two other points not at infinity (either one with multiplicity two, or two distinct points with multiplicity one). Hence, S equals R. That is, R is on the line through P and Q, and finally, P, Q and R are colinear.

On the other way, the addition law says us that adding three distinct points colinear on a curve E gives the point at infinity.

Exercise 7. Let C be the curve $u^2 + v^2 = c^2(1 + du^2v^2)$, as in Section 2.6.3. Show that the point $(c, 0)$ has order 4.

Solution 7. Here we can use the function `addition_affine_edwards` from the file `curve_arithmetic.py`.

```
from curve_arithmetic import addition_affine_edwards
#K.<u1,u2,v1,v2,c,d> = PolynomialRing(QQ)
#preparse("K.<u1,u2,v1,v2,c,d> = PolynomialRing(QQ)")
K = PolynomialRing(QQ, names=('u1', 'u2', 'v1', 'v2', 'c', 'd',));
(u1, u2, v1, v2, c, d,) = K._first_ngens(6)

u3, v3 = addition_affine_edwards((c,0), (c,0), c, d)
P2 = (u3, v3)
u4, v4 = addition_affine_edwards(P2, P2, c, d)
print("obtained 2*(c,0) = ({}, {})".format(u3,v3))
print("obtained 4*(c,0) = ({}, {})".format(u4,v4))
print("(0,c) is the neutral element for the group law on an Edwards form")
```

SageMath answer:

```
obtained 2*(c,0) = (0, -c)
obtained 4*(c,0) = (0, c)
(0,c) is the neutral element for the group law on an Edwards form
```

Exercise 8. Show that the method at the end of Section 2.2 actually computes kP. (Hint: Use induction on the length of the binary expansion of k. If $k = k_0 + 2k_1 + 4k_2 + \ldots + 2^\ell a_\ell$, assume the result holds for $k' = k_0 + 2k_1 + 4k_2 + \ldots + 2^{\ell-1}a_{\ell-1}$.)

Solution 8. This corresponds to the loop invariant on the slides.

Exercise 9. If $P = (x, y) \neq P_\infty$ is on the curve described by (2.1), then $-P$ is the other finite point of intersection of the curve and the vertical line through P. Show that $-P = (x, -a_1x - a_3 - y)$. (Hint: This involves solving a quadratic in y. Note that the sum of the roots of a monic quadratic polynomial equals the negative of the coefficient of the linear term.)

Solution 9.

```
# define a polynomial ring for the variables
QQx.<a1,a3,a2,a4,a6,l,x1,y1,x2,y2> = QQ[]
QQE.<X,Y> = QQx[]
E = Y^2 + a1*X*Y + a3*Y - X^3 - a2*X^2 - a4*X - a6
print("The negative of the point P(x1, y1) on E is:")
print("intersect the vertical at P (x1, y1) with the curve to get the negative of P that will be -P"
Eq = E(X=x1)
# y1 is a root, find the other one
print("Eq = E(X=x1); Eq % (Y-y1) = {} = {} mod E([x1, y1])".format(Eq % (Y - y1), (Eq % (Y - y1)) %
assert (Eq % (Y - y1)) % E([x1, y1]) == 0
Eq_ = Eq // (Y - y1); print("Eq // (Y - y1) = {}".format(Eq_))
y1_ = -Eq_.coefficient({Y:0})/Eq_.coefficient({Y:1}); print("finally, y1_ = {}".format(y1_))
print("-P = (x1, {})".format(y1_))
```

and SageMath answer is:

```
The negative of the point P(x1, y1) on E is:
intersect the vertical at P (x1, y1) with the curve to get the negative of P that will be -P
Eq = E([x1, Y]); Eq % (Y-y1) = -a2*x1^2 - x1^3 + a1*x1*y1 - a4*x1 + a3*y1 + y1^2 - a6 = 0 mod E([x1,
Eq // (Y - y1) = Y + a1*x1 + a3 + y1
finally, y1_ = -a1*x1 - a3 - y1
-P = (x1, -a1*x1 - a3 - y1)
```

Exercise 10 (Double-and-add)**.**

Solution 10. See the file `solutions_in_sage.sage` for solutions in short Weierstrass affine and Edwards affine coordinates, with lsb.

The msb form will be addressed in hand-in 3.

Exercise 11 (Points of order 2)**.**

Solution 11. The points of order 2 are characterized by $y = 0$, hence $x^3 + ax^2 + b = 0$. The coordinates are of the form $(x_i, 0)$ where x_i is a root of the cubic polynomial in x.

More precisely, we can assume $y^2 = x^3 + a_2x^2 + a_4x + a_6$. The points of order 2 are the $(x_i, 0)$ where x_0, x_1, x_2 are the three distinct roots of $x^3+a_2x^2+a_4x+a_6$. If x_0 is such a root, then $x^3+a_2x^2+a_4x+a_6 = (x-x_0)((x-x_0)^2+(\underbrace{a_2+3x_0}_{\frac{1}{2}f''(x_0)=a'\neq 0})(x-x_0)+\underbrace{3x_0^2+2a_2x_0+a_4}_{f'(x_0)=b'\neq 0} = x'(x'^2+a'x+b')$ and the other points of order 2 are $(x_1, 0)$ and $(x_2, 0)$ where x_1, x_2 are two distinct roots of $x'^2+a'x'+b'$, distinct from x_0. We can write $\Delta = a'^2-4b' = a_2^2-3a_4-(3x_0^2+2a_2x_0+a_4)$, the roots are $x_1 = (-a'-\sqrt{\Delta})/2 = (-a_2-3x_0-\sqrt{\Delta})/2$, $x_2 = (-a'+\sqrt{\Delta})/2 = (-a_2-3x_0+\sqrt{\Delta})/2$.

```
QQa.<a2,a4,a6,x0> = QQ[]
QQx.<x> = QQa[]
f = x^3 + a2*x^2 + a4*x + a6
t = x-x0
# f(x) = f(t+x0)
f(x+x0)
# x^3 + (a2 + 3*x0)*x^2 + (2*a2*x0 + 3*x0^2 + a4)*x + a2*x0^2 + x0^3 + a4*x0 + a6
```

Exercise 12 (Points of order 3)**.**

Solution 12. The points of order 3 are inflexion point (flex points). Let $E\colon y^2 = x^3+a_2x^2+a_4x+a_6 = f(x)$ and consider the function $g(x) = \sqrt{f(x)} = f^{1/2}(x)$. Its derivative is $g'(x) = 1/2f'(x)f^{-1/2}(x) = 1/2(3x^2 + 2a_2x + a_4)g^{-1}(x)$. Its second derivative is $h(x) = g''(x) = 1/2(f''(x)f^{-1/2}(x) + f'(x)(f^{-1/2}(x))') = 1/2(f''(x)f^{-1/2}(x) + f'(x)(-1/2f'(x)f^{-3/2}(x))) = 1/(4f^{-3/2}(x))(2f''(x) - f'^2(x))$. The numerator is $2f''(x) - f'^2(x) = 2(6x+2a_2)(x^3+a_2x^2+a_4x+a_6) - (3x^2+2a_2x+a_4)^2 = 4(3x^4+4a_2x^3+(a_2^2+3a_4)x^2 + (a_2a_4+3a_6)x+a_2a_6) - (9x^4+12a_2x^3+(4a_2^2+6a_4)x^2+4a_2a_4x+a_4^2) = 3x^4+4a_2x^3+6a_4x^2+12a_6x-a_4^2+4a_2a_6$. The points of order 3 are the points (x_i, y_i) where x_i is a root of $2f''(x) - f'^2(x)$ and y_i is such that (x_i, y_i) satisfies the curve equation. There are four distinct roots x_i, and there are two solutions $\pm y_i$ for each x_i, hence there are 8 points of order 3. In addition, there is $\mathcal{O}$, and there are 9 points of 3-torsion. We show that the roots x_i are distinct with SageMath: compute the resultant of h and h' and show that is is always non-zero.

```
QQa.<a2,a4,a6> = QQ[]
QQx.<x> = QQa[]
f = x^3 + a2*x^2 + a4*x + a6
g = sqrt(f)
((g.derivative(x)).derivative(x)).numerator().factor()
(4*a2*x^3 + 3*x^4 + 6*a4*x^2 - a4^2 + 4*a2*a6 + 12*a6*x)*sqrt(a2*x^2 + x^3 + a4*x + a6)
2*f.derivative(x).derivative(x)*f - f.derivative()^2
3*x^4 + 4*a2*x^3 + 6*a4*x^2 + 12*a6*x - a4^2 + 4*a2*a6
# does h have multiple roots? Compute the resultant of h and derivative h'
h.resultant(h.derivative(x)).factor()
# (-20736) * (-a2^2*a4^2 + 4*a2^3*a6 + 4*a4^3 - 18*a2*a4*a6 + 27*a6^2)^2
# can it be 0?
E = EllipticCurve([0,a2,0,a4,a6])
E.j_invariant().factor()
# (256) * (-a2^2 + 3*a4)^3 * (-a2^2*a4^2 + 4*a2^3*a6 + 4*a4^3 - 18*a2*a4*a6 + 27*a6^2)^-1
# the resultant has a square factor which is the denominator of the j-invariant, it is non-zero
# hence the resultant is always non-zero, and g has only simple roots
```