CSCI-762 | Assignment #5
Thomas Bottom | Due 19 March 2020

Exploring elliptic curves.
For edition 4 of the textbook, use chapter 7 (instead of 6), same exercise numbers, pages 306/307.

1. Solve exercise 6.13 page 278. Note that the answer in (c) must be a divisor of (a).

2. Solve exercise 6.14 page 279.

3. Solve exercise 6.15 page 279.

4. Solve exercise 6.16 page 279.

5. Proving associativity of point addition on elliptic curves is quite complicated. In this exercise you will do just a special case of it. Suppose that points P=(p1,p2) and Q=(q1,q2), p1 not equal to q1, are on an elliptic curve E (either real or modular). It is obvious that ((-P) + P) + Q = Q.
Prove that (-P) + (P + Q) = Q by
using geometric reasoning on the plane
using only algebraic transformations defining point addition

---

1. Exercise 7.13

The source code used to solve this question can be found in Appendix A.

(a) Determine the number of points on $\mathcal{E}$.
There are **72** points on $\mathcal{E}$.

(b) Show that $\mathcal{E}$ is not a cyclic group.
As can be seen in the **table below**, none of the points on the curve is a generator.

Point	Order	Point	Order	Point	Order
O	1	(35, 14)	12	(4, 5)	36
(27, 0)	2	(35, 57)	12	(4, 66)	36
(53, 0)	2	(52, 26)	12	(13, 26)	36
(62, 0)	2	(52, 45)	12	(13, 45)	36
(20, 5)	3	(66, 18)	12	(15, 9)	36
(20, 66)	3	(66, 53)	12	(15, 62)	36
(5, 4)	4	(1, 32)	18	(21, 3)	36
(5, 67)	4	(1, 39)	18	(21, 68)	36
(49, 24)	4	(6, 26)	18	(23, 19)	36
(49, 47)	4	(6, 45)	18	(23, 52)	36
(2, 31)	6	(12, 8)	18	(33, 1)	36
(2, 40)	6	(12, 63)	18	(33, 70)	36
(19, 27)	6	(22, 30)	18	(34, 23)	36
(19, 44)	6	(22, 41)	18	(34, 48)	36
(39, 32)	6	(25, 22)	18	(37, 33)	36
(39, 39)	6	(25, 49)	18	(37, 38)	36
(31, 32)	9	(58, 27)	18	(41, 7)	36
(31, 39)	9	(58, 44)	18	(41, 64)	36
(36, 12)	9	(61, 15)	18	(43, 22)	36
(36, 59)	9	(61, 56)	18	(43, 49)	36
(63, 17)	9	(65, 27)	18	(47, 5)	36
(63, 54)	9	(65, 44)	18	(47, 66)	36
(3, 22)	12	(69, 35)	18	(48, 11)	36
(3, 49)	12	(69, 36)	18	(48, 60)	36

(c) What is the maximum order of an element in $\mathcal{E}$? Find an element having this order.
The **maximum order is 36**. An example element with this order is the point **(4,5)**, as are all points in the rightmost column of the table above.

2. Exercise 6.14

Suppose that $p > 3$ is an odd prime, and $a, b \in \mathbb{Z}_p$. Further, suppose that the equation $x^3 + ax + b \equiv 0 \mod p$ has three distinct roots in $\mathbb{Z}_p$. Prove that the corresponding elliptic curve group $(\mathcal{E}, +)$ is not cyclic.

HINT Show that the points of order two generate a subgroup of $(\mathcal{E}, +)$ that is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

The only points P of order 2 are those of the form $2P = \mathcal{O}$, which are exclusively those points whose y coordinate equals 0. Take these three roots of the curve to be p_1, p_2, p_3 together with the identity element $\mathcal{O}$ as a subgroup of $(\mathcal{E}, +)$. This subgroup is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$, which is not cyclic, so $\mathbb{Z}_p$ containing this subgroup cannot be cyclic.

3. Exercise 6.15

Consider an elliptic curve $\mathcal{E}$ described by the formula $y^2 \equiv x^3 + ax + b \mod p$, where $4a^3 + 27b^2 \not\equiv 0 \mod p$ and $p > 3$ is prime.

(a) It is clear that a point $P = (x_1, y_1) \in \mathcal{E}$ has order 3 iff $2P = -P$. Use this fact to prove that, if $P = (x_1, y_1) \in \mathcal{E}$ has order 3, then

$$3x_1^4 + 6ax_1^2 + 12x_1b - a^2 \equiv 0 \mod p$$

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

$$x_1 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1 \qquad x_1 = x_2$$

$$3x_1 = (\frac{3x_1^2 + a}{2y_1})^2$$

$$3x_1 = \frac{9x_1^4 + 6ax^2 + a^2}{4y_1^2} \qquad \textit{do square}$$

$$12x_1y_1^2 = 9x_1^4 + 6ax^2 + a^2$$

$$12x^4 + 12ax^2 + 12bx = 9x_1^4 + 6ax^2 + a^2 \qquad \textit{substitute } \mathcal{E} \textit{ for } y^2$$

$$3x_1^4 + 6ax_1^2 + 12bx_1 - a^2 \equiv 0 \mod p$$

(b) Conclude from equation (6.7) that there are at most 8 points of order 3 on the elliptic curve $\mathcal{E}$.
The four roots of the quartic polynomial (6.7) form the x coordinate of 8 points on the curve of the form (x,y) and (x,-y).

(c) Using equation (6.7) determine all points of order 3 on the elliptic curve $y^2 \equiv x^3 + 34x \mod 73$
The four roots of $3x_1^4 + 204x_1^2 + -1156 \equiv 0 \mod 73$ are 1, 2, 71, 72
Solve $y^2 \equiv x^3 + 34x \mod 73$ with these four roots as x:

$x = 1, \quad y^2 = 35 \quad \mod 73y \quad = 20, 53$

$x = 2, \quad y^2 = 3 \quad \mod 73y \quad = 21, 52$

$x = 71, \quad y^2 = 70 \quad \mod 73y \quad = 17, 56$

$x = 72, \quad y^2 = 38 \quad \mod 76y \quad = 29, 44$

The 8 points of order 3 are (1,20), (1,53), (2,21), (2,52), (71,16), (71,56), (72,29), and (72,44).

4. Exercise 6.16

Suppose that $\mathcal{E}$ is an elliptic curve defined over $\mathbb{Z}_p$, where $p > 3$ is prime. Suppose that $\#\mathcal{E}$ is prime, $P \in \mathcal{E}$, and $P \neq \mathcal{O}$.

(a) Prove that the discrete logarithm $\log_p$(-P) = $\#\mathcal{E} - 1$.

If $\#\mathcal{E}$ is prime, then $\mathcal{E}$ is cyclic. $\log_p$(-P) must be $\#\mathcal{E} - 1$ because P+-P = $\mathcal{O} = \mathcal{E}P$.

(b) Describe how to compute $\#\mathcal{E}$ in time $O(p^{1/4})$ by using Hasse's bound on $\#\mathcal{E}$, together with a modification of SHANKS' ALGORITHM. Give a pseudocode description of the algorithm.

Hasse's bound asserts that $q + 1 - 2\sqrt{q} \leq \#\mathcal{E} \leq q + 1 + 2\sqrt{q}$ (where $q = p^n$ for p prime).

Modified Shanks:

// We do not know the order of the group, so set m to the square root of the upper bound

$m = \lceil \sqrt{q + 1 + 2\sqrt{q}} \rceil$

for(j = 1 to m)
    compute jP and store in a list L1

Sort the ordered pairs of L1, (j, jP) by the second coordinate

for(i = 1 to m)
    compute (-P) + im(-P) and store in a list L2

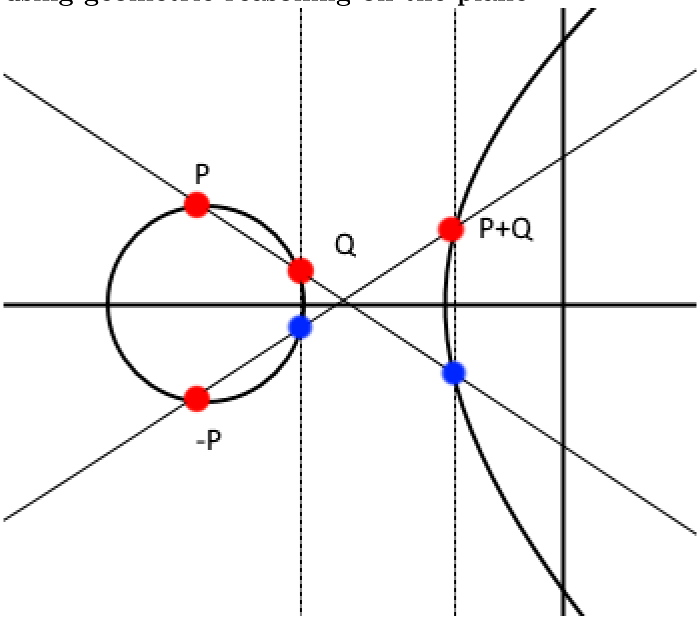
Sort the ordered pairs of L2, (i, (-P) + im(-P)) by the second coordinate

Find a pair (j, jP) and (i, (-P) + im(-P)) having identical second coordinates

$\#\mathcal{E}$ = im + j + 1

5. Proving associativity of point addition on elliptic curves is quite complicated. In this exercise you will do just a special case of it. Suppose that points P=(p1,p2) and Q=(q1,q2), p1 not equal to q1, are on an elliptic curve E (either real or modular). It is obvious that ((-P) + P) + Q = Q.
Prove that (-P) + (P + Q) = Q by

using geometric reasoning on the plane



using only algebraic transformations defining point addition

On page 293 of the textbook (4th Edition) the subtraction operation for points on an elliptic curve is defined as Q - P = Q + (-P). Using this definition we can subtract -P from both sides of $(-P) + (P + Q) = Q$ yielding:
$(P + Q) = Q - (-P) \rightarrow P + Q = Q + P$
We can immediately conclude that this relationship must hold for the case where $P = Q$ so the below proof is only concerned with showing that $(-P) + (P + Q) = Q$ for the case when $P \neq Q$.

We begin with the definitions of the coordinates P, -P, Q, and apply the standard formulae for point addition.

$$P = (x_1, y_1)$$
$$-P = (x_1, -y_1)$$
$$Q = (x_2, y_2)$$
$$P + Q = (x_3, y_3)$$
$$x_3 = \lambda_1^2 - x_1 - x_2$$
$$y_3 = \lambda_1(x_1 - x_3) - y_1$$
$$\lambda_1 = \frac{y_2 - y_1}{x_2 - x_1}$$
$$(-P) + (P + Q) = (x_4, y_4)$$
$$x_4 = \lambda_2^2 - x_1 - x_3$$
$$\lambda_2 = \frac{y_3 - (-y_1)}{x_3 - x_1} = \frac{y_1 + y_3}{x_3 - x_1} = \frac{y_1 + (\lambda_1(x_1 - x_3) - y_1)}{x_3 - x_1} = \lambda_1 \frac{x_1 - x_3}{x_3 - x_1} = -\lambda_1$$
$$x_4 = (-\lambda_1)^2 - x_1 - x_3 = \lambda_1^2 - x_1 - (\lambda_1^2 - x_1 - x_2) = x_2$$
$$y_4 = \lambda_2(x_1 - x_4) + y_1 = -(\lambda_1)(x_1 - x_2) + y_1 = -(y_2 - y_1)\frac{x_1 - x_2}{x_2 - x_1} + y_1 = -(y_2 - y_1)(-1) + y_1 = y2$$

$\boldsymbol{x_4 = x_2,\ y_4 = y_2 \rightarrow (-P) + (P + Q) = Q}$

A Source Code

```
# CSCI-762 Assignment 5
# Due 2020-03-19
# Thomas Bottom

# Global modulus used by the ecc module
# default 11, set via ecc.modulus = ...
# This is convenient for performing arithmetic with the Points defined
# below because they may be constructed with only their coordinates and
# no additional parameters need to be stored to use operator overrides
modulus = 11
a = 1
b = 6

# x^n mod m
def modpow(x, n, m):
    if n <  0: raise ValueError("n must be >= 0")
    if n == 0: return 1
    y = 1
    while n > 1:
        if n%2 == 0:
            x = (x*x) % m
            n = n / 2
        else:
            y = (x*y) % m
            n = n - 1
    return (x*y) % m

# assume modulus is prime, find inverse by exponentiation
def find_inverse(x, modulus):
    return modpow(x, modulus-2, modulus)

# returns True iff a^((p-1)/2) = 1 mod p, False otherwise
def quadratic_residue(a, p):
    return 1 == modpow(a, (p-1)/2, p)

class Point:
    def __init__(self, x, y, infinity=False):
        self.x = x
        self.y = y
        self.infinity=infinity

    def __eq__(self, other):
        return  (self.x == other.x and self.y == other.y) \
                  or (self.infinity and other.infinity)

    def __lt__(self, other):
        if self == other:      return False
        if self.x != other.x: return self.x < other.x
        else:                  return self.y < other.y
```

```
    def __add__(self, other):
        # handle identity case
        if self.infinity:
            return other
        elif other.infinity:
            return self

        # infinity case
        if self.x == other.x and self.y == (-other.y % modulus):
            return Point(0,0,True) # point at infinity
        # calculate slope
        elif self == other:
            slope = (((3*(self.x**2))+a) * find_inverse(2*self.y, modulus)) % modulus
        else:
            slope = ((other.y-self.y) * find_inverse(other.x-self.x, modulus)) % modulus
        # calculate (x3,y3)
        x3 = ((slope**2)-self.x-other.x) % modulus
        y3 = (slope*(self.x-x3)-self.y) % modulus
        return Point(x3,y3)

    def __str__(self):
        if self.infinity:
            return "O"
        else:
            return "(%d, %d)" % (self.x, self.y)

# Find all points on the globally defined curve and return them in a list
def find_points():
    assert(3 == modulus % 4)
    points = [Point(0,0,True)] # point at infinity
    for x in range(modulus):
        y2 = (modpow(x, 3, modulus) + (a*x) + b) % modulus
        if quadratic_residue(y2, modulus):
            y  = modpow(y2, ((modulus+1)/4), modulus)
            assert(modpow(y, 2, modulus) == y2)
            assert(modpow(-y%modulus, 2, modulus) == y2)
            points.append(Point(x, y))
            points.append(Point(x, -y%modulus))
        elif 0 == y2:
            points.append(Point(x,0))
    return points

# Find subgroup generated using point addition
def find_subgroup(point):
    subgroup = [Point(0,0,True)] # start with [O]
    alpha = point
    while not alpha.infinity:
        subgroup.append(alpha)
        alpha = alpha + point
    return subgroup

def find_order(point):
    return len(find_subgroup(point))
```

```python
if __name__ == "__main__":
    # 7.13 (a) find all points on y^2 = x^3 + x + 28 mod 71
    a = 1
    b = 28
    modulus = 71
    points = find_points()
    print "There are %d points on the curve" % len(points)

    # Do a little sanity check
    for p in points:
        sg = find_subgroup(p)
        for s in sg: assert(s in points)

    # 7.13 (b) show this is not a cyclic group
    print "Point & Order\\\\"
    orders = []
    for p in points:
        order = find_order(p)
        orders.append((p, order))

    orders.sort(key = lambda t : (t[1], t[0].x, t[0].y))
    for o in orders:
        print "{} & {} \\\\".format(o[0], o[1])
```