

Modern Cryptography Spring 2024

Exercises

⟨Ziqi WANG⟩
Wang05457@gtit.edu.cn

⟨2024.6⟩

Remark: I did NOT strictly follow the notions of the book. For example, for the notion Multiplicative Group, instead of Z_n^* , I use $U(n)$. And I used a lot of theorems probably not in the textbook but from my course of Abstract Algebra. To avoid of confusion, I will state the theorems I used as explicitly as possible.

Also, I used [SageMath](#) codes since sometimes Mathematica doesn't work for unknown reason.

Problem Nr. ⟨A.11⟩

Prove Theorem A.17 with the Principle of Inclusion and Exclusion (Thm. A.40) and the definition of the Euler function $\varphi(n)$.

Solution.

Theorem A.17

$$\phi(m) = m \prod_{p \text{ prime}, p|m} \left(1 - \frac{1}{p}\right).$$

Observation: Given a natural number n , then n can be factorized to a product of prime powers. So $\gcd(m, n)=1$ iff m is NOT divisible by any prime factors of n .

Suppose A_i is set of numbers that divisible by of p_i in the range from 1 to N , where p_i is prime factors of N . By our observation, A_i contains the elements we don't want to have, so we need to exclude them. The Euler function is defined as the number that is coprime to N in the range 1 to $N-1$. Thus, we can express the Euler function like this and simplify it using the inclusion and exclusion principle:

$$\begin{aligned}\phi(N) &= |\overline{A_1} \cap \overline{A_2} \dots \overline{A_k}| \quad (\text{By our observation}) \\ &= N - |A_1 \cup A_2 \dots \cup A_k| \quad (\text{De-Morgan's Law}) \\ &= N - \left(\frac{N}{p_1} + \frac{N}{p_2} + \dots + \frac{N}{p_k}\right) + \left(\frac{N}{p_1 p_2} + \frac{N}{p_1 p_3} + \dots\right) - \left(\frac{N}{p_1 p_2 p_3} + \frac{N}{p_1 p_2 p_4} + \dots\right) \quad (\text{I.E. principle})\end{aligned}$$

$$= N \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_k}\right)$$

Problem Nr. (B.05)

Find an element of order 12 in the group $(\mathbb{Z}_{13}^*, \times)$. Which powers of this element have order 12. Answer the same question for elements of order 6, 4, 3, 2 and 1.

Solution.

Observation: the order of Multiplicative Group $U(n)$ is exactly $\phi(n)$. By Lagrange's theorem, the order of elements in $U(n)$ divides the order of $U(n)$. Since 13 is prime then $\phi(13)=12$, also, $U(n)$ is cyclic and is isomorphic to Z_{12} . Let's enumerate the cases:

- 2 has order 12 (generator), since

$$2^1 \bmod 13 = 2$$

$$2^2 \bmod 13 = 4$$

$$2^3 \bmod 13 = 8$$

$$2^4 \bmod 13 = 3$$

$$2^5 \bmod 13 = 6$$

$$2^6 \bmod 13 = 12$$

$$2^7 \bmod 13 = 11$$

$$2^8 \bmod 13 = 9$$

$$2^9 \bmod 13 = 5$$

$$2^{10} \bmod 13 = 10$$

$$2^{11} \bmod 13 = 7$$

$$2^{12} \bmod 13 = 1$$

- 1 has order 1 is clear, since it is the identity element.
- 12 has order 2, since

$$12^1 \bmod 13 = 12$$

$$12^2 \bmod 13 = 1$$

- 3 has order 3, since

$$3^1 \bmod 13 = 3$$

$$3^2 \bmod 13 = 9$$

$$3^3 \bmod 13 = 1$$

- 5 has order 4, since

$$5^1 \bmod 13 = 5$$

$$5^2 \bmod 13 = 12$$

$$5^3 \bmod 13 = 8$$

$$5^4 \bmod 13 = 1$$

- 10 has order 6, since

$$10^1 \bmod 13 = 10$$

$$10^2 \bmod 13 = 9$$

$$10^3 \bmod 13 = 12$$

$$10^4 \bmod 13 = 3$$

$$10^5 \bmod 13 = 4$$

$$10^6 \bmod 13 = 1$$

Problem Nr. (B.09)

Make a log table of $\text{GF}(2)[x] / (1 + x^2 + x^5)$ (hint: x is a primitive element). Use this table to express $x^{10} + x^{20}$ as power of x .

Solution.

Preliminaries from Abstract Algebra:

1. $\text{Field} \subset \text{Euclidean Domain (ED)} \subset \text{Principle Ideal Domain (PID)} \subset \text{Unique Factorization Domain} \subset \text{Commutative ring with 1}$
2. For a ring R , R is field iff $R[x]$ is an ED.
3. Given a ring R be PID, let $I = aR$ be its ideal, then a is prime iff a is irreducible
4. Given a ring R be PID, let $I = aR$ be its ideal, then aR is maximal iff a is irreducible
5. Given a commutative ring R with 1, I is the maximal ideal of R iff R/I is a field.

Observation: note that $\text{GF}(2)$ is isomorphic to \mathbb{Z}_2 . Also note that \mathbb{Z}_n is a field iff n is prime. 2 is prime, so \mathbb{Z}_2 is a field. By prop 2, $\text{GF}(2)[x]$ is an ED. It is easy to verify that $1+x^2+x^5$ is irreducible and thus prime. Hence, the quotient is a field. More precisely, the quotient is

isomorphic to $Z_2[\alpha] = \{a + b\alpha + c\alpha^2 + d\alpha^3 + e\alpha^4 \mid a, b, c, d, e \in Z_2\}$, where α is the primitive element, satisfying $1 + \alpha^2 + \alpha^5 = 0$ and the cardinality of the quotient set is $2^5 = 32$.

Using the Mathematica codes:

```
f = x^5 + x^2 + 1;

(*Generate the results for PolynomialMod[x^n, f, Modulus -> 2] for n from 1 to 31*)
results = Table[PolynomialMod[x^n, f, Modulus -> 2], {n, 1, 31}];

(*make table*)
table = TableForm[Table[{n, results[[n]]}, {n, 1, 31}], TableHeadings -> {None, {"n", "PolynomialMod[x^n, f, Modulus -> 2]"}]];

(*Display table*)
table
```

n	PolynomialMod[x^n, f, Modulus -> 2]
1	x
2	x ²
3	x ³
4	x ⁴
5	1 + x ²
6	x + x ³
7	x ² + x ⁴
8	1 + x ² + x ³
9	x + x ³ + x ⁴
10	1 + x ⁴
11	1 + x + x ²
12	x + x ² + x ³
13	x ² + x ³ + x ⁴
14	1 + x ² + x ³ + x ⁴
15	1 + x + x ² + x ³ + x ⁴
16	1 + x + x ³ + x ⁴
17	1 + x + x ⁴
18	1 + x
19	x + x ²
20	x ² + x ³
21	x ³ + x ⁴
22	1 + x ² + x ⁴
23	1 + x + x ² + x ³
24	x + x ² + x ³ + x ⁴
25	1 + x ³ + x ⁴
26	1 + x + x ² + x ⁴
27	1 + x + x ³
28	x + x ² + x ⁴
29	1 + x ³
30	x + x ⁴
31	1

One can easily check by the table that x is a primitive element and $x^{14} = x^4 + x^3 + x^2 + 1$

So $x^{10} + x^{20} = x^4 + 1 + x^3 + x^2 = x^4 + x^3 + x^2 + 1 = x^{14}$

Problem Nr. <8.3>

Demonstrate the Special Case version of the Pohlig-Helmann algorithm, that computes logarithms in finite fields of size $q = 2^n + 1$, by evaluating $\log_3(142)$ in $GF(257)$.

Solution.

Goal: Given a finite field $GF(p) \cong Z_p$, where p is prime. Let α be generator of $GF(p)$. Every element of $GF(p)$ can be written as a power of α , namely, $c = \alpha^m$. Our goal is to find such m , which is unique modulo $p-1$.

Observation: consider $\phi(p) = p - 1$. If $p - 1 = st$, where s, t are coprime factors of $p-1$. Let $m = a + bs$, then $c = \alpha^{a+bs} \mod p$. By Fermat's theorem (special case of Euler's theorem, but enough for our problem), $c^t \equiv \alpha^{at+bst} \equiv \alpha^{at} \alpha^{bst} \equiv \alpha^{at} \mod p$. There we can find a , since it is a number between 0 and $p-1$ (possibly by brutal force). By modulo p both sides of $m = a + bs$, we know $m \equiv a \mod s$ (*).

Note that such one equation (*) is not enough to find an unique solution. By similar process, we can find $m \equiv a' \mod t$ (**), for $m = a' + b't$. By Chinese Remainder Theorem, the system of equations (*) and (**) admits unique solution.

Now, for the special case which our problem suppose, $2^n = p - 1$. Then $p-1$ can be factorized to the sum of powers of 2s. Namely, $m = m_0 + 2m_1 + 2^2m_2 + \dots + 2^{n-1}m_{n-1}$. We can still apply the raising powers method as upper, first determine m_0 by raising power to some value so that other terms vanish, then m_0 , term by term, until all knowns are solved.

Now back to our problem, we know, $257 = 2^8 + 1$ is prime, we want to find DLOG of $142 = 3^m \mod 257$.

First, we express the discrete logarithm in binary form:

$$m = m_0 + 2m_1 + 2^2m_2 + 2^3m_3 + 2^4m_4 + \dots + 2^7m_7$$

$$c = 142 = 3^{m_0+2m_1+2^2m_2+2^3m_3+2^4m_4+\dots+2^7m_7} \mod 257$$

Note that $2^7 = 128$, we want to find m_0 , so we want all other terms vanish. The coefficient of m_1 is 2, so $2 \times 128 = 256 = p - 1$. By raising powers of 128, all other terms vanish (By Fermat's). We then solve equation $142^{128} = 3^{128m_0} \mod 257$, obtain $m_0 = 1$. The procedure of textbook has essentially the same mechanism as above, but in a more formal, systematic manner. Here is the solution using the syntax of the book:

We find m_0 by evaluating $c^{(q-1)/2} = 142^{128} \mod 257 = -1$, so $m_0 = 1$

And, we get $c_1 = 142 \times 3^{-1} \mod 257 = 133$

In a similar way, we find m_1 by evaluating $c_1^{(q-1)/4} = 133^{64} \mod 257 = -1$, so $m_1 = 1$

And, we get $c_2 = 133 \times 3^{-2} \mod 257 = 129$

In a similar way, we find m_2 by evaluating $c_2^{(q-1)/8} = 129^{32} \bmod 257 = 1$, so $m_2 = 0$

And, we get $c_3 = 129 \times 3^0 \bmod 257 = 129$

In a similar way, we find m_3 by evaluating $c_3^{(q-1)/16} = 129^{16} \bmod 257 = 1$, so $m_3 = 0$

And, we get $c_4 = 129 \times 3^0 \bmod 257 = 129$

In a similar way, we find m_4 by evaluating $c_4^{(q-1)/32} = 129^8 \bmod 257 = -1$, so $m_4 = 1$

And, we get $c_5 = 129 \times 3^{-16} \bmod 257 = 16$

In a similar way, we find m_5 by evaluating $c_5^{(q-1)/64} = 16^4 \bmod 257 = 1$, so $m_5 = 0$

And, we get $c_6 = 16 \times 3^0 \bmod 257 = 16$

In a similar way, we find m_6 by evaluating $c_6^{(q-1)/128} = 16^2 \bmod 257 = -1$, so $m_6 = 1$

And, we get $c_7 = 16 \times 3^{-64} \bmod 257 = 256$

In a similar way, we find m_7 by evaluating $c_7^{(q-1)/256} = 256^1 \bmod 257 = -1$, so $m_7 = 1$

Finally, we get $m = (11010011)$ in binary, so **$m = 211$**

Check: $3^{211} \bmod 257 = 142$

PowerMod[3, 211, 257]

142

Hence, $m=211$ is indeed our desired DLOG.

Problem Nr. <8.7>

Check that $\alpha = 662$ is a primitive 2003-th root of unity in $\text{GF}(4007)$ (note that 4007 is a prime number). Let G be the multiplicative subgroup G of order 2003 in $\text{GF}(4007)$ generated by α . Check that 2124 is an element of G .

Determine $\log_{662} 2124$ by the Pollard- ρ method.

Solution. We get $662^{2003} \bmod 4007 = 1$, and here does not exist any positive integer $k < 2003$ such that $662^k = 1 \bmod 4007$, so 662 is a primitive 2003-th root of unity in $\text{GF}(4007)$.

If 2124 is an element of G , since α is a generator, then each element of G can be written as power of α . So $2124 = 662^x \bmod 4007$, which is a DLOG problem. we solve it by *Pollard* – ρ method, by modifying the Mathematica codes on textbook(p133-134).

The general idea of this solution is to generate two sequences: $x_i = \alpha^{a_i} c^{b_i}$ and $x_j = \alpha^{a_j} c^{b_j}$, such that satisfying a recurrence relation as follows:

We define a sequence $\{x_i\}_{i \geq 0}$ in $\text{GF}(q)$ recursively by $x_0 = 1$ and

$$x_{i+1} = f(x_i) = \begin{cases} (x_i^2 \bmod q), & \text{if } x_i \in G_0, \\ (c \cdot x_i \bmod q), & \text{if } x_i \in G_1, \\ (\alpha \cdot x_i \bmod q), & \text{if } x_i \in G_2. \end{cases}$$

We want to find an index such that x_i and x_j coincides. If so, by our construction, if $b_i \neq b_j$, we can conclude $m \equiv \frac{a_j - a_i}{b_i - b_j} \bmod p$. Else, we need to put $c' = c * \alpha$, and solve a new equation $c' = \alpha^{m+1}$.

Note that the only 4 parameters we need to modify are: alp(generator), c(variable of log), q(the prime), p(order of the subgroup)

The recurrence relation for the $\{x_i\}_{i \geq 0}$ sequence can be evaluated by means of the [Which](#) and [Mod](#) functions.

```
RecX[x_, alp_, c_, q_] :=
  Which[Mod[x, 3] == 0, Mod[x^2, q], Mod[x, 3] == 1, Mod[c * x, q], Mod[x, 3] == 2, Mod[alp * x, q]
```

The smallest index i , $i \geq 1$, satisfying $x_i = x_{2i}$ can quite easily be found with the help of the [While](#) function.

```
alp = 662; c = 2124; q = 4007;
x1 = RecX[1, alp, c, q];
x2 = RecX[x1, alp, c, q]; i = 1;
While[x1 != x2, x1 = RecX[x1, alp, c, q]; x2 = RecX[RecX[x2, alp, c, q], alp, c, q]; i = i + 1;
i
```

84

```
RecurrDef[{x_, a_, b_}] := Which[
  Mod[x, 3] == 0, {Mod[x^2, q], Mod[2 a, p], Mod[2 b, p]},
  Mod[x, 3] == 1, {Mod[c * x, q], a, Mod[b + 1, p]},
  Mod[x, 3] == 2, {Mod[alp * x, q], Mod[a + 1, p], b}]
```

```
alp = 662; c = 2124; q = 4007; p = 2003;
x1 = 1; a1 = 0; b1 = 0;
x2 = 1; a2 = 0; b2 = 0;
{x1, a1, b1} = RecurrDef[{x1, a1, b1}]; i = 1;
{x2, a2, b2} = RecurrDef[RecurrDef[{x2, a2, b2}]];
While[x1 != x2, {x1, a1, b1} = RecurrDef[{x1, a1, b1}];
  {x2, a2, b2} = RecurrDef[RecurrDef[{x2, a2, b2}]];
  i = i + 1;
Print["i=", i]
Print["\i\ (x\_i\)" =, x1, ", \i\ (a\_i\)" =, a1, ", \i\ (b\_i\)" =, b1];
Print["\i\ (x\_ (2. \i\))\)" =, x2, ", \i\ (a\_ (2. \i\))\)" =, a2, ", \i\ (b\_ (2. \i\))\)" =, b2];
```

i=84

x_i=121, a_i=856, b_i=41

x_{2.i}=121, a_{2.i}=1102, b_{2.i}=704

So the smallest index that the powers coincide is 84. We here extract the value of the two sequences $x_i = \alpha^{a_i} c^{b_i}$, $a_i = 856, a_j = 1102, b_i = 41, b_j = 704$, where $j = 2i$. We can check that $i=84$ is indeed the index such that $\alpha^{a_i} c^{b_i} = \alpha^{a_{2i}} c^{b_{2i}}$.

```
Mod[PowerMod[a1p, a1, q] * PowerMod[c, b1, q], q]
Mod[PowerMod[a1p, a2, q] * PowerMod[c, b2, q], q]
```

121

121

Since $b_i \neq b_j$, we can conclude $m \equiv \frac{1102-856}{41-704} \mod 4007$. Hence, $m=625$.

```
m = Mod[(a2 - a1) * PowerMod[b1 - b2, -1, p], p]
```

625

Check the answer.

```
PowerMod[a1p, m, q] == c
```

True

Therefore, the answer is 625.

Problem Nr. <9.6>

Give a complete factorization of $n = 110545695839248001$ by means of Pollard's ρ Algorithm.

Solution. Recall that Pollard's rho Algorithm takes an integer n as input, an prime factor p of n as output.

General idea:

1. Define two sequences, a 'slow' one: $a_i = (a_{i-1}^2 + 1) \mod n, a_0 = 1$ and a 'fast' one: $b_i = (b_{i-1}^2 + 1 \mod n)^2 \mod n, b_0 = 1$.
2. The two sequences have same starting point, but different growing speed. Since we are in a finite field, at some point, there exist an index such that two sequences have same value.
3. When the value coincides, $n \mid b - a$, let $d = \gcd(b - a, n)$, and such $d > 1$ is **very likely** to be a prime factor. Double check is needed.

The factoring algorithm is as following(textbook p161):


```

input : integer  $n$ .
put  $a = 1, b = 2$ .
do  $a \leftarrow (a^2 + 1) \bmod n$ ,
     $b \leftarrow ((b^2 + 1) \bmod n)^2 + 1 \bmod n$ 
until  $d = \gcd(b - a, n) > 1$ 
if  $d < n$  then  $d$  is a factor of  $n$ 
    else STOP

```

Pollard's ρ Method to Factor n

Figure 9.2

We got the factors are 230327045551 and 479951(is prime). (Check with Mathematica codes on page162)

```

n = 110 545 695 839 248 001;
a = 1; b = 2; d = GCD[b - a, n];
While[d == 1, a = Mod[a^2 + 1, n];
  b = Mod[(Mod[b^2 + 1, n])^2 + 1, n];
  d = GCD[b - a, n]
d

```

479 951

```
PrimeQ[479 951]
```

True

It turns out that the other factor 230327045551 is not prime, so the factorization is not complete.

```

a = n / 479 951
PrimeQ[a]

```

230 327 045 551

False

Input 230327045551 to factorize again, we got a new factor 479939(is prime).

```

n = 230 327 045 551;
a = 1; b = 2; d = GCD[b - a, n];
While[d == 1,      a = Mod[a2 + 1, n];
  b = Mod[(Mod[b2 + 1, n])2 + 1, n];
  d = GCD[b - a, n]]
d

```

479 939

```
PrimeQ[479 939]
```

True

Another factor is 479909 and it turns out that it is also prime. Hence the factorization is complete.

```

a = n / 479 939
PrimeQ[a]

```

479 909

True

In summary, $110545695839248001 = 230327045551 \times 479951 = 479939 \times 479909 \times 479951$.

Problem Nr. (9.10)

Suppose that Alice has sent the same secret message to B, C, D, E, and F by means of the RSA system. Let the public moduli of these people be given by $n_B = 324059$, $n_C = 324371$, $n_D = 326959$, $n_E = 324851$, and $n_F = 324899$. Assume that they all have the same public exponent $e = 5$.

Let the intercepted messages be given by $c_B = 68207$, $c_C = 96570$, $c_D = 251415$, $c_E = 273331$, resp. $c_F = 154351$.

Determine Alice's message (see Example 9.8).

Solution.

Recall the procedure:

1. First decide the Public Key (n_i, e) , in our case $e=5$.
2. Alice send message m , with encryption $c_i = m^5 \bmod n_i$
3. We then have a system of 5 equations each w.r.t. a receiver from B,C,D,E

To recover the message, it suffices to solve the following system of equation.

$$\begin{cases} m^5 \equiv c_B \pmod{n_B} \\ m^5 \equiv c_C \pmod{n_C} \\ m^5 \equiv c_D \pmod{n_D} \\ m^5 \equiv c_E \pmod{n_E} \\ m^5 \equiv c_F \pmod{n_F} \end{cases}$$

Let $N = n_B n_C n_D n_E n_F$, $x = m^e = (C_B M_B M_B^{-1} + C_C M_C M_C^{-1} + C_D M_D M_D^{-1} + C_E M_E M_E^{-1} + C_F M_F M_F^{-1}) \bmod N$, where $M_i = N/n_i$

By Chinese Remainder Theorem, $x \equiv m^5 \bmod N$ is the unique solution to the system of the equations. Using the codes from example 9.8:

```
nB=324059;nC=324371;nD=326959;nE=324851;nF=324899;
cB=68207;cC=96570;cD=251415;cE=273331;cF=154351;
mPowerFive=ChineseRemainderTheorem[{cB,cC,cD,cE,cF},{nB,nC,nD,nE,nF}]
```

408 526 801 936 602 069 773 048 832

```
m = (408 526 801 936 602 069 773 048 832)1/5
```

210 012

Finally, we recovered the message 210012.

Problem Nr. <10.2>

Find the intersection points over \mathbb{Z}_{31} of the lines $y = 4x + 20$ and $y = 4x + 21$ with the elliptic curve $y^2 = x^3 + 25x + 10$.

Solution. We use the Mathematica codes from example 10.2.

For the line $y=4x+20$: (11, 2), and (25, 27)

```
p = 31;
Clear[x];
ec = x3 + 25 x + 10;
il = 4 x + 20;
Factor[il2 - ec, Modulus -> p]
```

30 (6 + x) (20 + x)²

```
x = Mod[{ -6, -20, -20}, p]
y = Mod[4 * x + 20, p]
```

{25, 11, 11}

{27, 2, 2}

For the line $y=4x+21$: (1, 25), (17, 27), (29, 13)

```
p = 31;
Clear[x];
ec = x^3 + 25 x + 10;
il = 4 x + 21;
Factor[il^2 - ec, Modulus -> p]
```

30 (2 + x) (14 + x) (30 + x)

```
x = Mod[{ -2, -14, -30}, p]
y = Mod[4 * x + 21, p]
```

{29, 17, 1}

{13, 27, 25}

Problem Nr. (10.6)

Consider (again) the elliptic curve \mathcal{E} defined by $y^2 = x^3 + 11x^2 + 17x + 25$ over \mathbb{Z}_{31} .

Determine the orders of $P = \{27, 10\}$ and $Q = \{24, 28\}$. What can you conclude about the cardinality of \mathcal{E} (hint: use Theorem B.5)?

What is the cardinality of \mathcal{E} (hint: use Theorem 10.1)?

Construct a point of maximal order from P and Q .

Solution. Since we are over a finite field of modulo 31, the set of integer points on the elliptic curve has cardinality no more than 31.

Definition 10.2 *addition*

Let P be a point on an elliptic curve \mathcal{E} (so, it defined by (10.1)), with O as point at infinity. Then we define the sums

$$P + O = O + P = P.$$

Further, let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on \mathcal{E} , both not O . Then the *sum* $P_1 + P_2$ is defined by

- i) $P_3 = -Q$ if $x_1 \neq x_2$.
Here, Q is the third point of intersection of \mathcal{E} with of the line \mathcal{L} through (x_1, y_1) and (x_2, y_2) .
- ii) $P_3 = -Q$ if $P_1 = P_2$ and the tangent line through P is a single tangent.
Here, Q is the third point of intersection of \mathcal{E} with the tangent \mathcal{L} through P .
- iii) $P_3 = -P_1$ if $P_1 = P_2$ and the tangent line through P is a double tangent.
- iv) $P_3 = O$ if $P_1 = -P_2$.

Theorem 10.2

The points on an elliptic curve \mathcal{E} together with the addition defined in [Definition 10.2](#) form an additive group. The zero element is given by O .

By 10.2, $G=(\mathcal{E}, +)$ is a finite additive group. Since P and Q are elements of this group, the order of P and Q are no more than 31. We can enumerate all possible multiples of each element and find the first coefficient s.t. it goes to zero (the point at infinity by 10.2). We use SageMath codes to find their orders.

```

SageMath 10.3 Trusted Kernel is idle (halt...) CPU RAM
In [1]: F = FiniteField(31)
        E = EllipticCurve(F, [0, 11, 0, 17, 25])
        print(E)
        G1 = E(27,10)
        G2 = E(24,28)
        print(G1.order())
        print(G2.order())
        for i in range(2,31):
            if i* G1 == G1:
                print('G1 ',i)
            if i* G2 == G2:
                print('G2 ',i)

Out[1]: Elliptic Curve defined by y^2 = x^3 + 11*x^2 + 17*x + 25 over Finite Field of size 31
5
6
G1 6
G2 7
G1 11
G2 13
G1 16
G2 19
G1 21
G2 25
G1 26

```

It turns out that the order of P is 5 and the order of Q is 6.

Theorem B.5

Let (G, \cdot) be a finite group of order n . Then every subgroup (H, \cdot) of (G, \cdot) has an order dividing n . Also every element a , $a \neq e$, in G has an order dividing n .

Since we found two elements of G whose orders are 5, 6. Let n denote the order of G . By Theorem 10.5, $5 \mid n$ and $6 \mid n$. Since 5 and 6 are coprime, then 30 is their least common multiple. Note that the point constructed by P, Q is a linear combination of P and Q . So the maximal order of elements constructed by P and Q is a multiple of 30.

And with the Theorem 10.1: $21 \leq N \leq 43$

Theorem 10.1 *Hasse*

Let N be the number of points on an elliptic curve over $\text{GF}(q)$. Then

$$|N - (q + 1)| \leq 2\sqrt{q}$$

We check if $N=30$, the inequality is SAT. Hence, the cardinality of the integer points on $\mathcal{E} \bmod 31$ is 30 and the maximal order of elements constructed by P and Q is exactly 30.

Problem Nr. {15.3}

Construct a $(7, 4)$ -threshold scheme over the finite field $GF(16) = GF(2)[a]/(a^4 + a + 1)$ (see Theorem B.15).

What are the shares of the participants for secret $S = (1, 0, 1, 1)$ which stands for the field element a^{13} ? Show in detail how participants 2, 4, 5, 7 recover S .

Solution. By similar arguments as problem B.09, it is easy to show the quotient is isomorphic to a finite field F , which is generated by a basis $\{1, \alpha, \alpha^2, \alpha^3\}$, where α SAT $\alpha^4 + \alpha + 1 = 0$. In our case, $F = GF(16)$. One can easily check $\alpha^{13} = \alpha^3 + \alpha + 1$.

In a $(7, 4)$ -threshold scheme, the secret is shared using a polynomial of degree 3.

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3, \text{ where } a_0 \text{ is the secret and } a_i \in GF(16).$$

We need to distribute the 'shares' to 7 people, so that more than 4 people (including 4) together can recover the secret, or else fail (since there are 4 coefficients in the polynomial). Also note that in our case, $a_0 = \alpha^3 + \alpha + 1$.

To share the secret, for each participant $x_i \in GF(16)$, we associate it to $p(x_i)$, which stands for the 'shares'. Each element can be represented as a linear combination of the basis.

To recover the secret, any 4 participants can use their shares and apply Lagrange interpolation to reconstruct the polynomial $p(x)$.

Verify with sagemath codes as follows:

```
# Part 1: Construct GF(2^4)
```

```
print("Part 1: Constructing the finite field GF(2^4)")
```

```
F.<a> = GF(2^4) # Define the finite field GF(2^4) with the element 'a'
```

```
print("Finite field GF(2^4):", F)
```

```
S = a^13 # Define the secret as a^13 in GF(2^4)
```

```
print("Secret S = a^13:", S)
```

```
print()
```

```
# Part 2: Construct the polynomial
```

```
print("Part 2: Constructing the polynomial f(x)")
```

```
PR.<x> = PolynomialRing(F) # Define a polynomial ring over the field F with variable 'x'
```

```
# Create a random polynomial f(x) of degree 3 with the constant term S
```

```

f = F.random_element()*x^3 + F.random_element()*x^2 + F.random_element()*x + S
print("Polynomial f(x):", f)
assert f(0) == S # Verify that the constant term of the polynomial is S
print("Verification passed: f(0) == S")
print()

```

Part 3: Share the secret

```

print("Part 3: Sharing the secret among participants")
participants = ["pad"]
for i in range(7):
    xi = F.from_integer(i) # Convert integer i to an element of the finite field F
    share = (xi, f(xi)) # Compute the share by evaluating f(x) at xi
    participants.append(share)
    print(f"Participant {i+1}: x = {xi}, share = {f(xi)}")

```

Part 4: Recover the secret

```

print("\nPart 4: Recovering the secret using shares from participants 2, 4, 5, and 7")
print("Participants used for recovery:")
for i in [2, 4, 5, 7]:
    print(f"Participant {i}: x = {participants[i][0]}, share = {participants[i][1]}")

```

Construct the Lagrange interpolation polynomial using the shares

```

recover_f = PR.lagrange_polynomial([participants[2], participants[4], participants[5],
participants[7]])
print("\nRecovered Polynomial:")
print(recover_f)

```

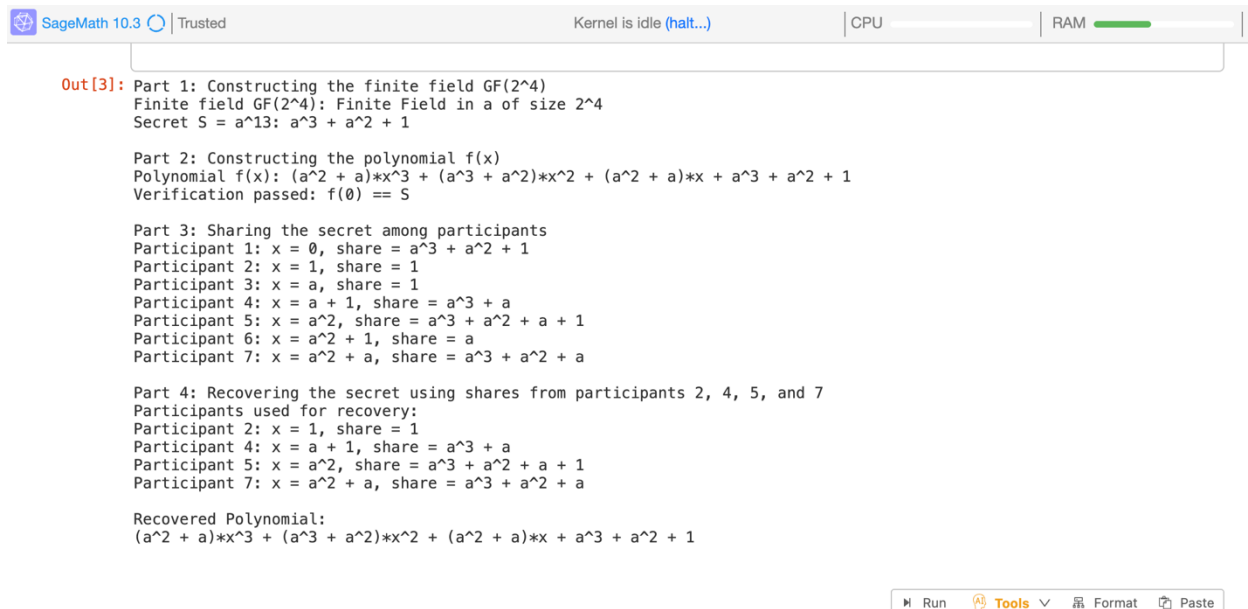
```
# Evaluate the recovered polynomial at x = 0 to get the secret

secret = recover_f(0)

print(f"\nRecovered Secret: {secret}")

assert secret == S # Verify that the recovered secret matches the original secret

print("\nAssertion passed: The recovered secret matches the original secret.")
```



```
SageMath 10.3 | Trusted | Kernel is idle (halt...) | CPU | RAM

Out[3]: Part 1: Constructing the finite field GF(2^4)
Finite field GF(2^4): Finite Field in a of size 2^4
Secret S = a^13: a^3 + a^2 + 1

Part 2: Constructing the polynomial f(x)
Polynomial f(x): (a^2 + a)*x^3 + (a^3 + a^2)*x^2 + (a^2 + a)*x + a^3 + a^2 + 1
Verification passed: f(0) == S

Part 3: Sharing the secret among participants
Participant 1: x = 0, share = a^3 + a^2 + 1
Participant 2: x = 1, share = 1
Participant 3: x = a, share = 1
Participant 4: x = a + 1, share = a^3 + a
Participant 5: x = a^2, share = a^3 + a^2 + a + 1
Participant 6: x = a^2 + 1, share = a
Participant 7: x = a^2 + a, share = a^3 + a^2 + a

Part 4: Recovering the secret using shares from participants 2, 4, 5, and 7
Participants used for recovery:
Participant 2: x = 1, share = 1
Participant 4: x = a + 1, share = a^3 + a
Participant 5: x = a^2, share = a^3 + a^2 + a + 1
Participant 7: x = a^2 + a, share = a^3 + a^2 + a

Recovered Polynomial:
(a^2 + a)*x^3 + (a^3 + a^2)*x^2 + (a^2 + a)*x + a^3 + a^2 + 1
```

Since the program continues executing without any interruption. This means the secret was successfully recovered using the shares of the participants, and it matches the original secret.