Питон, теория чисел

Хашин С.И.
http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html

Ивановский университет

.

Питон, классы на примере эллиптических кривых

.

Иваново-2023

План

Math

elliptic_curve

ГОСТ

Внутри

Сложение

Умножение

main

Определение эллиптической кривой

Эллиптические кривые применяются в криптографии и для выработки электронной подписи.
Пусть p — простое число > 3. Эллиптической кривой E, определённой над конечным простым полем $\mathbb{Z}_p$, называется множество пар чисел (x, y), $x, y \in \mathbb{Z}_p$, удовлетворяющих тождеству

$$y^2 = x^3 + ax + b \ (\mod p),$$

где $a, b \in \mathbb{Z}_p$ и $4a^3 + 27b^2$ не сравнимо с нулем по модулю p.
Дополним кривую E еще одной "бесконечно удаленной" точкой, не имеющей координат, которую будем обозначать символом "0" (она будет нейтральным элементом аддитивной группы).

Сложение точек на эллиптической кривой

На множестве всех точек эллиптической кривой введем операцию сложения, которую будем обозначать знаком "+" по следующим правилам.

- $P + 0 = 0 + P$ для любой точки P кривой E, то есть бесконечно удаленная точка "0" – нейтральный элемент нашей группы.
- Пусть точки P_1 имеет координаты (x, y), точка P_2 имеет координаты $(x, -y)$. Тогда $P_1 + P_2 = 0$, то есть точки с координатами (x, y) и $(x, -y)$ – противоположны друг другу (их сумма равна 0).

Сложение точек на эллиптической кривой 2

- Пусть точки P_1 имеет координаты (x_1, y_1), точка P_2 имеет координаты (x_2, y_2), причем $x_1 \neq x_2$. Тогда точка $P_3 = P_1 + P_2$ имеет координаты (x_3, y_3), причем

$$x_3 = \lambda^2 - x_1 - x_2 \mod p$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \mod p$$

где

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

- И, наконец, последний случай: пусть точка P имеет координаты (x, y). Тогда $P + P = (x_3, y_3)$, где

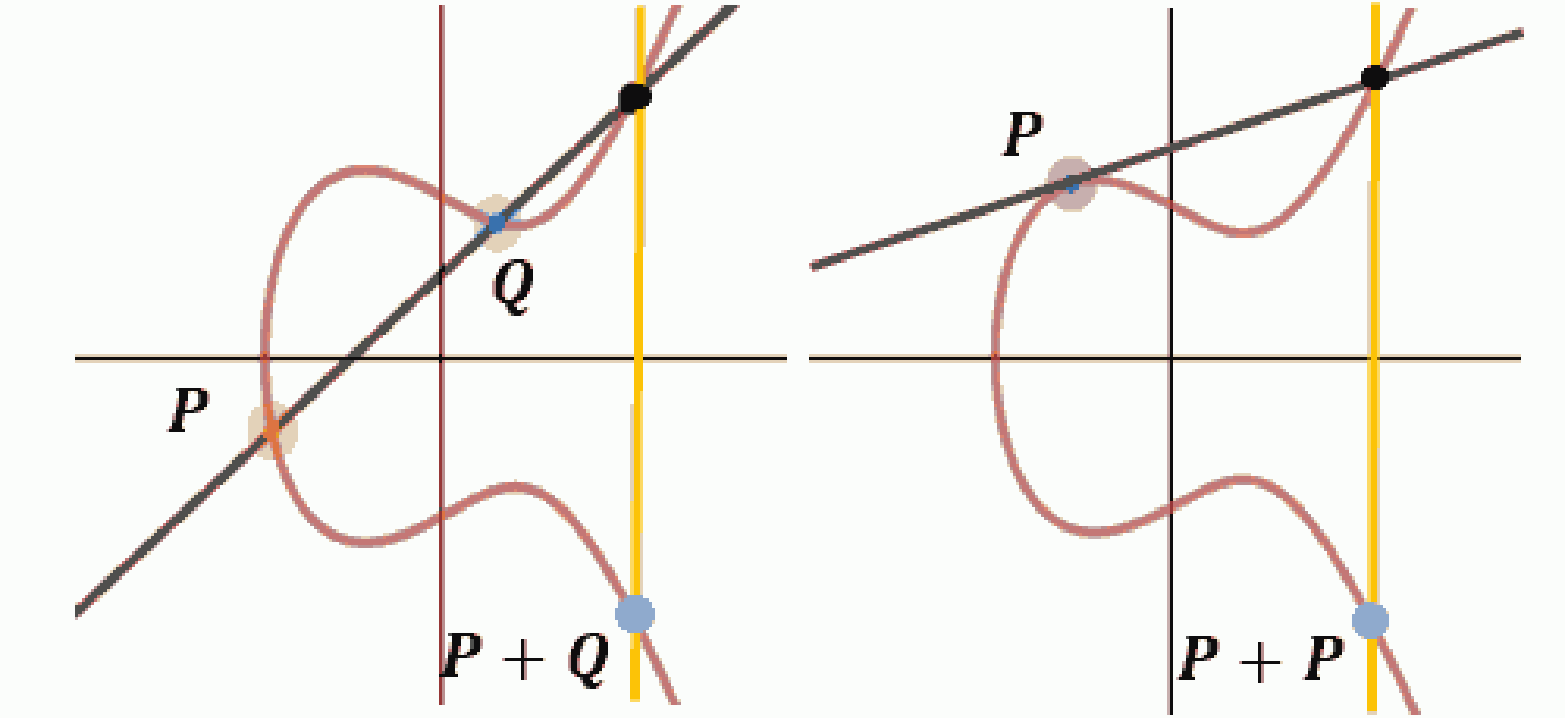
$$x_3 = \lambda^2 - 2x \mod p$$

$$y_3 = \lambda(x - x_3) - y \mod p$$

и

$$\lambda = \frac{3x^2 + a}{2y}.$$

Сложение точек на эллиптической кривой 3



Сложение точек на эллиптической кривой 4

Точки эллиптической кривой образуют абелеву группу относительно сложения, то есть:

- $P + 0 = 0 + P$ для любой точки P кривой E, то есть бесконечно удаленная точка "0" – нейтральный элемент нашей группы.
- $P + Q = Q + P$ — сложение коммутативно.
- $(P + Q) + R = P + (Q + R)$ — сложение ассоциативно.
- Для натурального k можно определить кратное точки: $kP = P + \cdots + P$ (k раз).
- Для любой точки P некоторое её кратное равно 0: $kP = 0$. Наименьшее такое $k > 0$ называется порядком точки.

Модуль elliptic_curve

Как использовать?

```
from elliptic_curve import ell

ell.set_101()
print('Ell.curve, p=', ell.p, 'a=', ell.a, 'b=', ell.b)
> Ell.curve, p= 101 a= 2 b= 21  Параметры класса!
print('P0=', ell.P0, 'P0 order=', ell.q)
>P0= (0, 18) P0 order= 107      Параметры класса!

A = ell(77,17)  # создали точку с координатами (77,17)
print('A=', A)  # A= (77, 17)

A = ell(77,18)  # Ошибка!
Exception: left is not right
y**2 = 21, f(x) = 87
```

Модуль elliptic_curve

```
A = ell(77,17)  # создали точку с координатами (77,17)
print('A=', A)
> A= (77, 17)
print('A+A=', A+A, '2*A=', 2*A, 'A*2=', A*2)
> A+A= (54, 72) 2*A= (54, 72) A*2= (54, 72)

print(7*A, (7*A).isZero())
> (30, 66) False
print(107*A, (107*A).isZero())
> (0) True

print(7*A, (7*A).isZero())
> (30, 66) False
print(107*A, (107*A).isZero())
> (0) True
```

Реальный пример из ГОСТ Р 34.10-2018

```
ell.set_gost()
print('Ell.curve, p=',ell.p,'\na=',ell.a)
> Ell.curve, p= 57896044618658097711785492504343953926\
  6634992332820282019728792003956564821041
> a= 7
print('b=',ell.b)
> b=43308876546767276905765904595650931995942111794451\
    039583252968842033849580414
print('P0=', ell.P0)
> P0= (2, 40189740565390375033354494229370597756357397\
          38990554508069097936521343156628 0)
print('P0 order=', ell.q)
>P0 order= 57896044618658097711785492504343953927082934\
          58372545062238097359213763106961 9
```

elliptic_curve.py

```
class ell:
  ''' Точки на эллиптической кривой y**2 = x**3 + a*x + b mo
  # p,a,b - переменные класса, а не экземпляра!
  # К ним обращаемся ell.p, ell.a, ell.b
  def __init__(self,x,y):#создание объекта (точки на кривой)
    if x == 0 and y == 0:
      self.x = self.y =-1 # невозможное значение
    else:
      left = pow(y, 2, ell.p)
      right = (pow(x, 3) + ell.a * x + ell.b) % ell.p
      if left != right:
          raise Exception(f"left is not right\n y**2 "\
                          f"= {left}, f(x) = {right}")
      self.x = x % ell.p
      self.y = y % ell.p
```

Определение класса

```
def __str__(self):
    ''' строковое представление объекта '''
    if self.x==-1: return ' (0) '
    return f'({self.x}, {self.y})'

# метод класса, применяется не к объекту (A.set_101()),
# а ко всему классу в целом (ell.set_101())
@classmethod
def set_101(self):
    ell.p = 101
    ell.a = 2
    ell.b = 21
    ell.P0 = ell(0,18)
    ell.q = 107
```

Сложение точек

```
def __add__(self, Q):
  ''' сложение точек self+Q '''
  if self.x==-1: return Q
  if Q.x==-1:    return self
  if self.x == Q.x and (self.y + Q.y)%ell.p ==0:
    return ell(0, 0)
  if self.x == Q.x:
    # lam = (3x*x+a)/(2*y) mod p
    lam=mod_div(3*self.x*self.x+ ell.a, 2*self.y, ell.p)
  else:
    # lam = (y2-y1)/(x2-x1) mod p
    lam = mod_div(Q.y-self.y, Q.x-self.x, ell.p )
  x3 = lam*lam - self.x - Q.x
  y3 = lam*(self.x - x3) - self.y
  return ell(x3, y3)
```

Умножение точек

```
def mult(self, k):
  ''' кратное: складываем self сам с собою k раз '''
  if k<0: raise Exception("ell.mult: k=%d"%k)
  res = ell(0,0) # результат здесь будет
  p2 = 1         # степень двойки, начинаем с нулевой
  ap2 = self     # p2*self
  while k>0:
    if k&1>0: res += ap2
    k //=2
    ap2 += ap2
  return res

def __mul__(self, k):   return self.mult(k)
def __rmul__(self, k):  return self.mult(k)
```

main

```
def test_0():   # простая проверка
  ell.set_101()
  print('Ell.curve, p=', ell.p, 'a=', ell.a, 'b=', ell.b)
  print('P0=', ell.P0, 'P0 order=', ell.q)

ell.set_101() # кривая по умолчанию, p=101, a=1, b=4
if __name__== "__main__":
   test_0()
```