

정수론 HW5

20011759 박수민

1. α 가 대수적인 수 (algebraic number)일 때, $\frac{3}{2}\alpha$ 도 대수적인 수임을 보이시오

α 가 대수적인 수 이므로 $f(x) = a_0 + a_1x^1 + \dots + a_nx^n$, $f(\alpha) = 0$ 인 임의의 $f(x)$ 가 존재한다.

$g(x) = a_0 + a_1\frac{2}{3}x^1 + a_2\left(\frac{2}{3}\right)^2x^2 + \dots + a_n\left(\frac{2}{3}\right)^nx^n$ 이라 하면, $g\left(\frac{3}{2}\alpha\right) = 0$ 이고 $g(x)$ 의 모든 계수는 유리수 이므로 정의에 의해 $\frac{3}{2}\alpha$ 는 대수적인 수가 된다

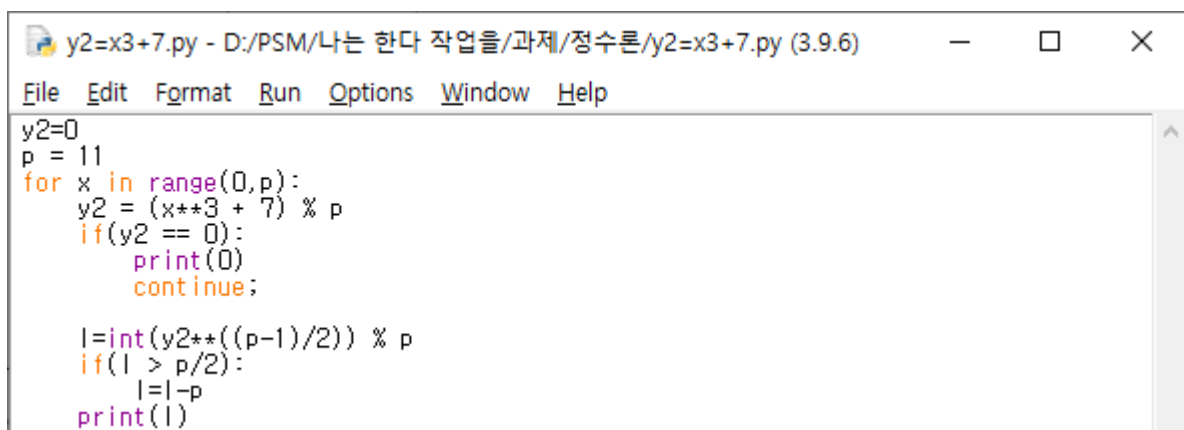
2. $y^2 = x^3 + 7$ 의 법 $p = 11$ 에 대한 합동식의 해의 개수를 구하시오

우선 법 11에 대해 모든 x 에 대해 각 y^2 값을 보면 다음과 같다

$$\begin{aligned}x &\equiv 0, y^2 \equiv 7 \\x &\equiv 1, y^2 \equiv 8 \\x &\equiv 2, y^2 \equiv 4 \\x &\equiv 3, y^2 \equiv 1 \\x &\equiv 4, y^2 \equiv 5 \\x &\equiv 5, y^2 \equiv 0 \\x &\equiv 6, y^2 \equiv 3 \\x &\equiv 7, y^2 \equiv 9 \\x &\equiv 8, y^2 \equiv 2 \\x &\equiv 9, y^2 \equiv 10 \\x &\equiv 10, y^2 \equiv 6\end{aligned}$$

이제 법 11에 대한 각 y^2 값들의 이차잉여 값을 확인하면 된다.

이는 파이썬으로 계산하였다.



```
y2=x3+7.py - D:/PSM/나는 한다 작업을/과제/정수론/y2=x3+7.py (3.9.6)
File Edit Format Run Options Window Help
y2=0
p = 11
for x in range(0,p):
    y2 = (x**3 + 7) % p
    if(y2 == 0):
        print(0)
        continue;

    l=int(y2**((p-1)/2)) % p
    if(l > p/2):
        l=-p
    print(l)
```

효율성은 고려하지 않고 11정도면 적당히 작은 수라 오일러 판정법으로 계산하였다. p가 큰 수라면 이차상호법칙들을 이용하여 개선하면 된다. 계산 결과는 다음과 같다

```
===== RESTART: D:/PSM/나는 한다 작업을/과제/정수론/y2=x3+7.py =====
0 : -1
1 : -1
2 : 1
3 : 1
4 : 1
5 : 0
6 : 1
7 : 1
8 : -1
9 : -1
10 : -1
>>>
```

Ln: 68 Col: 4

-1인 경우는 이차비잉여로 해가 존재하지 않고, 1인 경우 존재하나 음수, 양수 각 2개씩이고 0인 경우는 근이 단순 0으로 1개만 있다. 1이 5개, 0이 1개이므로 $5 \times 2 + 1 = 11$, 11개가 존재한다.

3. $\left| \frac{a}{b} - \sqrt{19} \right| < \frac{1}{b^3}$ 을 만족하는 유리수 $\frac{a}{b}$ 는 $b < 10$ 임을 보이고 유리수 $\frac{a}{b}$ 를 하나 구하시오.

$f(x) = x^2 - 19 = (x - \sqrt{19})(x + \sqrt{19})$, $g(x) = (x + \sqrt{19})$ 라 하자,

$\left| g\left(\frac{a}{b}\right) \right| \leq \left| \frac{a}{b} \right| + \sqrt{19} \leq (\sqrt{19} + 1) + \sqrt{19}$ 에서 $\left| g\left(\frac{a}{b}\right) \right| \leq 2\sqrt{19} + 1$ 이고 $\left| f\left(\frac{a}{b}\right) \right| \geq \frac{1}{b^2}$ 이므로

$\frac{1}{b^2} \leq \left| f\left(\frac{a}{b}\right) \right| \leq \frac{2\sqrt{19}+1}{b^3}$, 이므로 $b \leq 2\sqrt{19} + 1$ 이다. 이는 약 9.7178 정도이므로 $b < 10$ 이다.

위에서 $b < 10$ 임을 구했으므로 적당한 숫자를 대입하여 a도 구할 수 있다.

$b = 3$ 일 때, $a = 13$, $\left| \frac{a}{b} - \sqrt{19} \right| = 0.02556 \dots$, $\frac{1}{b^3} = 0.03703 \dots$ 으로 $\left| \frac{a}{b} - \sqrt{19} \right| < \frac{1}{b^3}$ 를 만족한다.

4. 유리수에서 정의된 타원곡선 $y^2 = x^3 - 2$ 의 점 $P = (3, 5)$ 에 대하여 13P를 구하시오.

이에 대한 계산은 타원곡선상의 점에 대한 덧셈군과 스칼라 곱을 파이썬으로 구현하였다.

정정 : 13P를 구하시오 -> 4P를 구하시오

```
타원곡선.py - D:\PSM\나는 한다 작업을\과제\정수론\타원곡선.py (3.9.6)
File Edit Format Run Options Window Help
from fractions import Fraction

def add_points(P, Q):
    if P == Q:
        s = Fraction((3 * P[0] ** 2), (2 * P[1]))
    else:
        s = Fraction((Q[1] - P[1]), (Q[0] - P[0]))

    x3 = s**2 - P[0] - Q[0]
    y3 = s * (x3 - P[0]) + P[1]
    y3 *= -1
    return x3, y3

def mult_points(P, m):
    if(m == 1):
        return P
    if(m == 2):
        return add_points(P, P)

    mbit1 = m % 2;
    m = m >> 1;
    if mbit1 == 1:
        return add_points(P, mult_points(mult_points(P, m), 2))
    else:
        return mult_points(mult_points(P, m), 2);

P = (3, 5)
P4 = mult_points(P, 4)
print("4P =", P4[0], ", ", P4[1])
```

```
IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\PSM\나는 한다 작업을\과제\정수론\타원곡선.py =====
4P = 2340922881/58675600 , 113259286337279/449455096000
>>>
```

5. 유한체 Z_{1013} 에서 정의된 타원곡선 $y^2 = x^3 - 4x + 4$ 위의 점 $P = (2, 2)$ 에 대하여 $100P$ 를 구하시오.

4번과 크게 다를 게 없다 다만 법 1013 상에서 계산을 해야하며 기울기를 계산할 때 나누는 부분은 유리수체 상에서의 나눗셈이 아닌 정수체에서의 곱셈 역원을 곱하여 계산해야한다. 이부분은 유클리드 알고리즘을 응용한 확장 알고리즘을 사용했으며 파이썬으로 구현하였다

```
타원곡선mod.py - D:/PSM/나는 한다 작업을/과제/정수론/타원곡선mod.py ...
File Edit Format Run Options Window Help

def extended_gcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, x, y = extended_gcd(b % a, a)
        return (g, y - (b // a) * x, x)

def mod_inverse(a, m):
    g, x, _ = extended_gcd(a, m)
    if g != 1:
        raise Exception('역원이 존재하지 않습니다.')
    else:
        return x % m

def add_points(P, Q, mod):
    if P == Q:
        s = (3 * P[0] ** 2 - 4) * mod_inverse(2 * P[1], mod)
    else:
        s = (Q[1] - P[1]) * mod_inverse(Q[0] - P[0], mod)

    x3 = s**2 - P[0] - Q[0]
    y3 = s * (x3 - P[0]) + P[1]
    y3 *= -1
    x3 %= mod
    y3 %= mod
    return x3, y3

def mult_points(P, m, mod):
    if (m == 1):
        return P
    if (m == 2):
        return add_points(P, P, mod)

    mbit1 = m % 2;
    m = m >> 1
    if mbit1 == 1:
        return add_points(P, mult_points(mult_points(P, m, mod), 2, mod), mod)
    else:
        return mult_points(mult_points(P, m, mod), 2, mod);

P = (3, 5)
mod = 1013
print("100P =", mult_points(P, 100, mod))
```

```
IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help

Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/PSM/나는 한다 작업을/과제/정수론/타원곡선mod.py =
=====
100P = (1001, 37)
>>> |
```