MA7010 – Number Theory for Cryptography - Assignment 3

Ajeesh Thattukunnel Vijayan

January 11th 2024

1 Introduction

Name	a	b	c	d	Q6
Ajeesh	2929	20953	500657	23861	ii

Table 1: Input Numbers For Ajeesh

2 Answers

1. For the number a given to you on page 1 answer the following:

   a. Show that a can be written as the sum of squares in two different ways

   Answer:

   $a = 2929$
   Let $u = 5, v = 2, w = 10, x = 1$
   Brahmagupta Identity helps us to write this numbers in two different sum of squares form
   $(u^2 + v^2)(w^2 + x^2) = (uw - vx)^2 + (ux + vw)^2$
   $= (uw + vx)^2 + (ux - vw)^2$
   $\therefore (uw - vx)^2 + (ux + vw)^2 = (5 \times 10 - 2 \times 1)^2 + (5 \times 1 + 2 \times 10)^2 = 48^2 + 25^2$
   and $(uw + vx)^2 + (ux - vw)^2 = (5 \times 10 + 2 \times 1)^2 + (5 \times 1 - 2 \times 10)^2 = 52^2 + (-15)^2$
   Hence the two different sum of squares forms of 2929 are:
   $2929 = 48^2 + 25^2$
   $2929 = 52^2 + 15^2$ ■

   b. Hence apply Euler's method to factorise a.

   Answer:

   From 1.(a) above, we have
   $2929 = 48^2 + 25^2 = a^2 + b^2$
   $2929 = 52^2 + 15^2 = c^2 + d^2$
   $\therefore a^2 + b^2 = c^2 + d^2$
   $\implies a^2 - c^2 = b^2 - d^2$
   $\implies (a - c)(a + c) = (b - d)(b + d)$

Let $k = gcd(a - c, b + d)$ and $h = gcd(a + c, b + d)$

$\therefore k = 2$ and $h = 20$

$\implies a - c = k.l \implies 4 = 2 \times l \implies l = 2$

$b - d = k.m \implies 10 = 2 \times m \implies m = 5$

$a + c = h.m' \implies 100 = 20 \times m' \implies m' = 5$

$b + d = h.l' \implies 40 = 20 \times l' \implies l' = 2$

$\therefore n = 2929 = ((\frac{k}{2})^2 + (\frac{h}{2})^2)(l^2 + m^2)$

$= (1^2 + 10^2)(2^2 + 5^2) = (101 \times 29)$ ■

2. (a) Take the number b assigned to you on page 1 and apply the gcd method to find its smallest factor.

Answer:

$b = 20953$

Set $k = \lfloor \sqrt{b} \rfloor = 144$

$P_0 = \prod_{primes<144} p = 100146466505991900675092331316499400573663346532004330 90$

$gcd(100146466505991900675092331316499400573663346532004330 90, 20953) = 23,$

which is the smallest factor of 20953. ■

The Maple worksheet for this exercise is below:

```
> p0 := 1
```

$$p0 := 1 \quad (1)$$

```
> k := 2;
```

$$k := 2 \quad (2)$$

```
> while k < 144 do p0 := p0 · k;
k := nextprime(k);
od;
```

$p0 := 2$

$k := 3$

$p0 := 6$

$k := 5$

$p0 := 30$

$k := 7$

$p0 := 210$

$k := 11$

$p0 := 2310$

$k := 13$

$p0 := 30030$

$k := 17$

$p0 := 510510$

$k := 19$

$p0 := 9699690$

$k := 23$

$p0 := 223092870$

$k := 29$

$p0 := 6469693230$

$k := 31$

$p0 := 200560490130$

$k := 37$

$p0 := 7420738134810$

$k := 41$

$p0 := 304250263527210$

$k := 43$

$p0 := 13082761331670030$

$k := 47$

$p0 := 614889782588491410$

$k := 53$

$p0 := 32589158477190044730$

$k := 59$

$p0 := 1922760350154212639070$

$k := 61$

$p0 := 117288381359406970983270$

$k := 67$

$p0 := 7858321551080267055879090$

$k := 71$

$p0 := 557940830126698960967415390$

$k := 73$

$p0 := 40729680599249024150621323470$

$k := 79$

$p0 := 3217644767340672907899084554130$

$k := 83$

$p0 := 267064515689275851355624017992790$

$k := 89$

$p0 := 2376874189634555077065053760135831 0$

$k := 97$

$p0 := 230556796394551842475310214733175607 0$

$k := 101$

$p0 := 232862364358497360900063316880507363070$

$k := 103$

$p0 := 239848235289252281727065216386922583962 10$

$k := 107$

$p0 := 256637611759499941447959781534007164839447 0$

$k := 109$

$p0 := 279734996817854936178276161872067809674997230$

$k := 113$

$p0 := 316100546404176077881452062915436624932746869 90$

$k := 127$

$p0 := 401447693933303618909444119902604513664588524773 0$

$k := 131$

$p0 := 525896479052627740771371797072411912900610967452630$

$k := 137$

$p0 := 720478176302100004856779361989204320673837025410103 10$

$k := 139$

$p0 := 100146466505991900675092331316499400573663346532004330 90$

$k := 149$ (3)

> $igcd(100146466505991900675092331316499400573663346532004330 90, 20953)$

23 (4)

>

(b) What is the value of P_0 that you need to guarantee finding a factor given that b is composite?

Answer: From the Maple worksheet from 2(a), the P_0 value that guarantee a minimum factor is the product of all primes upto 23 inclusive of 23. Hence the required $P0 = 223092870$ ■

3. Take the number c assigned to you on page and use the $p-1$ method to find one factor of c. You may assume that $c-1$ factorises into primes of $size < 100$.

Answer:

$N = 500657$
Set $L = [2, 3, 2, 5, 7, 2, 3, 11, 13, 2, 17, 19, 23, 5, 3, 29, 31, 2, 37, 41,$
$43, 47, 49, 53, 59, 61, 2, 67, 71, 73, 79, 3, 83, 89, 97]$
$a = 3$

num	**a**	$b = a^{num} \pmod N$	**gcd(b - 1, N)**
2	3	9	1
3	9	729	1
2	729	30784	1
5	30784	131942	1
7	131942	388833	1
2	388833	197744	1
3	197744	261270	1
11	261270	18971	1
13	18971	275059	1
2	275059	170269	1
17	170269	138053	1
19	138053	381008	1
23	381008	73825	1
5	73825	438038	101

Table 2: Pollard's P-1 Factorisation

Hence the smallest factor of 500657 is 101 ■

The Rust code for generating the above result is below.

```
pub fn pollards_p_1 (n: &BigInt , mut a: BigInt ) {
    if miller_rabin_primality (n) {
        println! ("{} is a prime ", &n);
        return ;
    }
    // This list need to be generated dynamically .
    // Static it is for now .
    let prime_powers : Vec<BigInt> = vec! [
    BigInt :: from (2 u64 ),
    BigInt :: from (3 u64 ),
    BigInt :: from (2 u64 ),
    BigInt :: from (5 u64 ),
    BigInt :: from (7 u64 ),
    BigInt :: from (2 u64 ),
    BigInt :: from (3 u64 ),
    BigInt :: from (11 u64 ),
    BigInt :: from (13 u64 ),
    BigInt :: from (2 u64 ),
    BigInt :: from (17 u64 ),
    BigInt :: from (19 u64 ),
    BigInt :: from (23 u64 ),
    BigInt :: from (5 u64 ),
    BigInt :: from (3 u64 ),
    BigInt :: from (29 u64 ),
    BigInt :: from (31 u64 ),
```

```
                    BigInt::from(2u64),
                    BigInt::from(37u64),
                    BigInt::from(41u64),
                    BigInt::from(43u64),
                    BigInt::from(47u64),
                    BigInt::from(49u64),
                    BigInt::from(53u64),
                    BigInt::from(59u64),
                    BigInt::from(61u64),
                    BigInt::from(2u64),
                    BigInt::from(67u64),
                    BigInt::from(71u64),
                    BigInt::from(73u64),
                    BigInt::from(79u64),
                    BigInt::from(3u64),
                    BigInt::from(83u64),
                    BigInt::from(89u64),
                    BigInt::from(97u64),
                ];

                for num in prime_powers.iter() {
                    let b = modular_pow(&a, num, n);
                    let gcd = n.gcd_euclid(&(&b - BigInt::one()));
                    println!("{:>10} {:>10} {:>10} {:>5}",
                            num, &a, &b, gcd);
                    if &gcd > &BigInt::one() {
                        break;
                    }
                    a = b;
                }
            }
```

Listing 1: Pollard's P-1 Factorisation

4. Take the number d assigned to you on page and use the $p-1$ method to find both factors of d. You may NOT use the maple procedure provided in weblearn.

   Answer: $d = 23861$

   i.

   $$i = 2, a = 2, b = a^i \pmod{d} = 2^2 \pmod{23861} = 4,$$
   $$23861 = 7953 \times 3 + 2, 3 = 1 \times 2 + 1 \implies gcd = 1$$

   ii.

   $$i = 3, a = 4, b = 4^3 \pmod{23861} = 64,$$
   $$23861 = 378 \times 63 + 47, 63 = 1 \times 47 + 16, 47 = 2 \times 16 + 15, 16 = 1 \times 15 + 1$$
   $$\implies gcd = 1$$

   iii.

   $$i = 2, a = 64, b = 64^2 \pmod{23861} = 4096,$$
   $$23861 = 5 \times 4095 + 3386, 4095 = 1 \times 3386 + 709, 3386 = 4 \times 709 + 550,$$
   $$709 = 1 \times 550 + 159$$
   $$550 = 3 \times 159 + 73, 159 = 2 \times 73 + 13, 73 = 5 \times 13 + 8,$$
   $$13 = 1 \times 8 + 5, 8 = 1 \times 5 + 3,$$
   $$5 = 1 \times 3 + 2, 3 = 1 \times 2 + 1 \implies gcd = 1$$

iv.

$i = 5, a = 4096, b = 4096^5 \pmod{23861} = 17634$
Calculating gcd takes too much time. Hence I used the gcd calc and modular power calc I develo
$gcd(23861, 17633) = 1$

v.

$i = 7, a = 17634, b = 17634^7 \pmod{23861} = 8861$
$gcd(23861, 17633) = 1$

vi.

$i = 2, a = 8861, b = 8861^2 \pmod{23861} = 14631$
$gcd(23861, 14630) = 1$

vii.

$i = 3, a = 14631, b = 14631^3 \pmod{23861} = 7152$
$gcd(23861, 7151) = 1$

viii.

$i = 11, a = 7152, b = 17643^11 \pmod{23861} = 10163$
$gcd(23861, 10162) = 1$

ix.

$i = 13, a = 10163, b = 10163^13 \pmod{23861} = 2401$
$gcd(23861, 2400) = 1$

x.

$i = 2, a = 2401, b = 2401^2 \pmod{23861} = 14300$
$gcd(23861, 14299) = 1$

xi.

$i = 17, a = 14300, b = 14300^17 \pmod{23861} = 10052$
$gcd(23861, 10051) = 1$

xii.

$i = 19, a = 10052, b = 10052^19 \pmod{23861} = 23584$
$gcd(23861, 23583) = 1$

xiii.

$i = 23, a = 23584, b = 23584^23 \pmod{23861} = 10069$
$gcd(23861, 10068) = 1$

xiv.

$i = 5, a = 10069, b = 10069^5 \pmod{23861} = 9432$
$gcd(23861, 9431) = 1$

xv.

$$i = 3, a = 9432, b = 9432^3 \pmod{23861} = 20788$$
$$gcd(23861, 20787) = 1$$

xvi.

$$i = 29, a = 20788, b = 20788^29 \pmod{23861} = 15228$$
$$gcd(23861, 15227) = 1$$

xvii.

$$i = 31, a = 15228, b = 15228^31 \pmod{23861} = 20741$$
$$gcd(23861, 20740) = 1$$

xviii.

$$i = 2, a = 20741, b = 20741^2 \pmod{23861} = 22973$$
$$gcd(23861, 22972) = 1$$

xix.

$$i = 37, a = 20741, b = 20741^37 \pmod{23861} = 3792$$
$$gcd(23861, 3791) = 223 > 1 \implies 223|23861 \text{and is one of the factors}$$
$$23861/223 = 107 \implies 107 \text{ is the other factor}$$
$$23861 = 107 \times 223$$ ■

5. Take the same number d and now factorise using the Quadratic Sieve method. You may use Maple commands included in the week 10 workshop folder.

Answer: $d = 23861$

Step 1. We use the polynomial $y(x) = x^2 - d$ to find the B-smooth numbers. We initialise $x = \sqrt{500657}$ to start the sieve. Calculate the square root of d.
$a = \lfloor\sqrt{23861}\rfloor = 154$
We then $y(x), y(x+1), y(x+2), ...$

Step 2. Calculate the factor base. Euler's criteria for odd primes to determine whether a number is a quadratic residue or not is in action here. We got the below Factor Base:
$\{2, 5, 19, 29\}$. We added -1 to it to consider negative values too. Hence the Factor Base became $\{-1, 2, 5, 19, 29\}$

Step 3. Calculate $y(x)$ and sieve the B-smooth numbers. Here we set $B = 4$.

Step 4. the table below shows the sieved numbers and the corresponding exponents matrix (I wrote some Rust code to 2 generate these values.)

x	y(x + a)	-1	2	5	19	29	37	41
-63	91	1	2	1	1	0	0	1
-55	99	1	2	1	1	0	1	0
-25	129	1	2	1	2	0	0	0
-18	136	1	0	1	0	1	1	0
-10	144	1	0	5	0	0	0	0
0	154	0	0	1	0	1	0	0
1	155	0	2	0	0	0	0	1
2	156	0	0	2	1	0	0	0
11	165	0	2	0	0	2	0	0
19	173	0	2	0	0	0	1	1
40	194	0	0	2	1	1	0	0
127	281	0	2	2	1	1	0	0
165	319	0	2	2	1	0	0	1
Selected Rows' sum		2	2	6	2	0	0	0
Values for v		1	1	3	1	0	0	0

Table 3: Quadratic Sieve Factorisation

Step 5. We calculate v and u as follows:

$$
\begin{aligned}
u &= 129 \times 144 \pmod{23861} \\
&= 18576 \\
v &= -1 \times 2 \times 5^3 \times 19 \pmod{23861} \\
&= 19111 \\
u^2 &\equiv v^2 \pmod{23861} \implies gcd(d, u - v) and gcd(d, u + v) are factors of d \\
gcd(23861, 19111 - 18576) &= 107 \\
gcd(23861, 19111 + 18576) &= 223 \\
\therefore 23861 &= 101 \times 223
\end{aligned}
$$

■

Step 6. The Rust code for generating the above result is below.

```
pub fn prepare_matrix (n: &BigInt ) {
    let mut primes = vec![BigInt::from(2u64)];
    let a = n.sqrt();
    println!("Square Root of {} = {}", n, a);

    let mut factor_base = vec![
    BigInt::from(2u64),
    BigInt::from(5u64),
    BigInt::from(7u64),
    BigInt::from(11u64),
    BigInt::from(13u64),
    BigInt::from(17u64),
    BigInt::from(19u64),
    BigInt::from(23u64),
    BigInt::from(29u64),
    BigInt::from(31u64),
    BigInt::from(37u64),
    BigInt::from(41u64),
    ];

```

```
    println!("Legendre Symbol is calculated using Euler's criteria:");
    println!("If n^(p-1)/2 (mod p) = 1,
                then (n/p) = 1, else (n/p) = -1");
    factor_base
       .retain(|x| modular_pow(n, &((x - 1) / BigInt::from(2u64)),
                 x) == BigInt::one());
    // factor_base.insert(0, BigInt::from(-1i32));
    println!("The calculated Factor Base is: {:?}", &factor_base);
    let mut y_x: Vec<BigInt> = Vec::new();
    // start = sqrt(n) - 100, end = sqrt(n) + 200
    // These values should be dynamic
    let start = a.clone() - BigInt::from(100u64);
    let end = a.clone() + BigInt::from(200u64);

    let mut m_by_n: Vec<Vec<i32>> = Vec::new();
    for i in range_inclusive(start, end) {
        let x = &i - &a;
        y_x.push(x.clone());
        // y(x) = (x + a)^2 - n
        let mut y = &i * &i - n;
        if y.sign() == Sign::Minus {
            y = -1 * y;
        }
        let p_factors = y.prime_factors(&mut primes).clone();
        let p_factors_map: HashMap<BigInt, i32> = p_factors
        .iter()
        .cloned()
        .map(|(p, e)| (p, e as i32))
        .collect();
        let distinct_factors = p_factors
        .iter()
        .map(|x| x.0.clone())
        .collect::<Vec<BigInt>>();
        let set1: HashSet<BigInt> = factor_base
            .iter().cloned().collect();
        let set2: HashSet<BigInt> = distinct_factors
            .iter().cloned().collect();

        if set2.is_subset(&set1) {
            // println!("{} {} {:?}", i - &a, &y, p_factors);

            let mut one_by_n: Vec<i32> = Vec::new();
            for base in factor_base.iter() {
                if set2.contains(&base) {
                    let e = p_factors_map.get(&base).unwrap();
                    one_by_n.push(e.clone());
                } else {
                    one_by_n.push(0);
                }
            }

            if x.sign() == Sign::Minus {
                one_by_n.insert(0, 1);
            } else {
                one_by_n.insert(0, 0);
            }
            m_by_n.push(one_by_n.clone());
            println!("{:>5} {:>5} {:?}", x, i, one_by_n);
        }
    }
}
```

```

```

Listing 2: Quadratic Sieve

We can use the below command to generate the desired matrix:

```
./target/debug/nt-assignments quadratic-sieve -n 23861
Square Root of 23861 = 154
Legendre Symbol is calculated using Euler's criteria :
If n^(p-1)/2 (mod p) = 1, then (n/p) = 1, else (n/p) = -1
The  calculated  Factor  Base  is: [2, 5, 19, 29, 37, 41]
-63     91   [1, 2, 1, 1, 0, 0, 1]
-55     99   [1, 2, 1, 1, 0, 1, 0]
-25    129   [1, 2, 1, 2, 0, 0, 0]
-18    136   [1, 0, 1, 0, 1, 1, 0]
-10    144   [1, 0, 5, 0, 0, 0, 0]
  0    154   [0, 0, 1, 0, 1, 0, 0]
  1    155   [0, 2, 0, 0, 0, 0, 1]
  2    156   [0, 0, 2, 1, 0, 0, 0]
 11    165   [0, 2, 0, 0, 2, 0, 0]
 19    173   [0, 2, 0, 0, 0, 1, 1]
 40    194   [0, 0, 2, 1, 1, 0, 0]
127    281   [0, 2, 2, 1, 1, 0, 0]
165    319   [0, 2, 2, 1, 0, 0, 1]

```

Listing 3: Quadratic Sieve Matrix Generation

6. The Maple worksheet in Weblearn for this assignment shows part of an attempt to factorise $N = 9263 = 59 * 157$ using the Number Field Sieve. In this we claim the following:

   i. $A = [0, 1, 0]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 2$

   ii. $B = [-1, -1, 0]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 3$

   iii. $C = [1, 0, 1]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 5$

   iv. $D = [1, 1, -1]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 11$

   v. $E = [1, -2, 0]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 17$

   vi. $F = [3, 0, -1]$ is a prime(irreducible) element of $Z(\sqrt[3]{-2})$ with $norm = 23$

   We also derive a 48×15 matrix R consisting of values of a and b such that $a + 21b$ can be factorised using small primes and $[a, b, 0]$ can be factorised in $Z(\sqrt[3]{-2})$ using just the primes $\{A, B, C, D, E, F\}$ and a unit element $U = [1, 1, 0]$.

   a. Prove the statement above allocated to you on page 1 using the definition of a norm.

      Answer: Norm is defined as the product of all the conjugates of the minimum polynomial in the field. i,e. in the For the polynomial $f(x) = x^3 + 2$, the roots are $\theta = \{\sqrt[3]{-2}, \theta.\frac{(-1+i\sqrt{3})}{2}, \theta.\frac{(-1-i\sqrt{3})}{2}\}$. Hence the algebraic integers in $Z(\sqrt[3]{-2})$ are of the form $a + b\theta + c\theta^2$. We represent these integers as $[a, b, c]$. The norm is defined as:

      $$N[a, b, c] = a^3 - 2b^3 + 4c^3 + 6abc. \quad \text{hence in our case, } [a, b, c] = [-1, -1, 0] \implies N([a, b, c]) = -1 +$$

      Since the $norm = 1$, the given element $B = [-1, -1, 0]$ is a *unit* in $Z(\sqrt[3]{-2})$, not a prime. $\square$

   b. Show that rows 23, 37, 41 and 45 of the matrix form a linearly dependent set modulo 2

      Answer: The rows 23, 37, 41, 45 in matrix form is:

$$A_{4\times 15} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 5 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 9 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$A \pmod 2$ gives, $A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$

Performing row operations on A to find the Row Echelon form:
R2 = R2 + R1,
R3 = R3 + R1,
R4 = R4 + R1:

$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$

Swap R2 and R4

$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$

R3 = R3 + R2

$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$

R3 and R4 are the same, R4 is a linear combination of the row R4. Hence the matrix becomes:

$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Hence the $rank(A) = 3$ (the number of leading 1's in the B) which is less than the number of rows in A and that implies A that the row vectors that form the matrix A are linearly dependent. □

c. Hence find an equation of the form u2 – v2 such that u2 – v2 and N have a common factor of 59 thus factorising N. You may use the Maple procedure for multiplication in $Z(\sqrt[3]{-2})$ to help you find u and v.

Answer:

$N = 9263 = 21^3 + 2$

An elementβ is of the form$(a + b\theta + c\theta^2) \in \mathbb{Z}(\sqrt[3]{-2})$

The Algebraic Factor Base given is $= \{U = [1, 1, 0], A = [0, 1, 0], B = [-1, 1, 0], C = [1, 0, 1], D = [1$

The Rational Factor Base given is $= \{-1, 2, 3, 5, 7, 11, 13\}$

The below table has the values selected for calculating u and v:

a	**b**	**a+21b**	**-1**	**2**	**3**	**5**	**7**	**11**	**13**	**-1**	**U**	**A**	**B**	**C**	**D**	**E**	**F**
0	2	42	0	1	1	0	1	0	0	1	0	4	0	0	0	0	0
6	0	6	0	1	1	0	0	0	0	1	1	3	3	0	0	0	0
7	3	70	0	1	0	1	1	0	0	1	1	0	0	0	0	2	0
-8	8	160	0	5	0	1	0	0	0	1	0	9	1	0	0	0	0
Sum Of Rows			0	10	2	2	2	0	0	4	2	16	4	0	0	2	0

Table 4: Number Field Sieve Factorisation

Calculating the v value by multiplying the rational factor bases with halved

values from the Sum Row: $v = 2^5 \times 3 \times 5 \times 7 = 1680$
Calculating the u value by multiplying the algebraic factor bases with halved values from the Sum Row:

$$\begin{aligned}\beta =& [a, b, c] \\ =& U \times A^8 \times B^2 \times E \\ =& [1, 1, 0] \times [0, 1, 0]^8 \times [-1, 1, 0] \times [-1, 1, 0] \times [1, -2, 0]\end{aligned}$$

We get value for u after substituting for a, b, c in $a + b\theta + c\theta^2$ where $\theta = 21$.
$u = -8 + 21 \times -8 + 21^2 \times -20 = -8996$
$u^2 = 6448 \pmod{9263}, u^2 = 6448 \pmod{9263}$
And we get $v^2 \equiv u^2 \pmod{9263}$

Calculating gcd to find the factors:

$$gcd(9263, 1680 + 8996) = 157$$
$$gcd(9263, 1680 - 8996) = 59$$

Hence, N is factorised as $N = 9263 = 59 \times 157$ ■

The Maple calculation performed is given below:

```
>
> norm1 := proc(a, b, c) global k; k := a^3 - 2·b^3 + 4·c^3 + 6·a·b·c; end;
```

$$-10258 \quad (5)$$

```
> U := [1, 1, 0]; A := [0, 1, 0]; B := [-1, 1, 0]; C := [1, 0, 1]; D1 := [1, 1, -1]; E := [1, -2, 0]; F := [3, 0, -1];
```

$$U := [1, 1, 0]$$

$$A := [0, 1, 0]$$

$$B := [-1, 1, 0]$$

$$C := [1, 0, 1]$$

$$D1 := [1, 1, -1]$$

$$E := [1, -2, 0]$$

$$F := [3, 0, -1] \quad (6)$$

```
> mult1 := proc(x, y); mult2(x[1], x[2], x[3], y[1], y[2], y[3]) ; end;
> mult4 := proc()global L; L := []; for i from 1 to nargs
do L := [op(L), args[i]]; od;
if nops(L) = 2 then
mult1(op(1, L), op(2, L)) else k := [mult1(op(1, L), op(2, L))]; L
:= subsop(1 = NULL, L); L := subsop(1 = NULL, L); L := [k, op(L)]; mult4(op(L)); fi; end;
> mult4(U, U, B, E);
```

$$1, 5, 3 \quad (7)$$

```
> H := mult4(U, A, A, A);
```

$$H := -2, -2, 0 \tag{8}$$

>

> $H := [-2, -2, 0]$;

$$H := [-2, -2, 0] \tag{9}$$

> $mult4(H, A, A, A)$;

$$4, 4, 0 \tag{10}$$

> $H := [4, 4, 0]$;

$$H := [4, 4, 0] \tag{11}$$

> $mult4(H, A, A, B)$;

$$0, -8, -4 \tag{12}$$

> $H := [0, -8, -4]$;

$$H := [0, -8, -4] \tag{13}$$

> $mult4(U, H, B, E)$;

$$32, -16, -28 \tag{14}$$

> $M := mult1(0, -8, -4, -1, 1, 0)$;

$$M := 0, 0, 0 \tag{15}$$

> $mult1(0, 0, 0, 1, -2, 0)$;

$$0, 0, 0 \tag{16}$$

>

U*Aˆ8*Bˆ2*E = $[1, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [0, 1, 0] \cdot [-1, 1, 0] \cdot [-1, 1, 0] \cdot [1, -2, 0]$;

>

> $mult1(U, A)$;

$$0, 1, 1 \tag{17}$$

> $UA := [0, 1, 1]$;

$$UA := [0, 1, 1] \tag{18}$$

> $mult1(UA, A)$;

$$-2, 0, 1 \tag{19}$$

> $UAA := [-2, 0, 1]$;

$$UAA := [-2, 0, 1] \tag{20}$$

> $mult1(UAA, A)$;

$$-2, -2, 0 \quad (21)$$

> $UAAA := [-2, -2, 0]$;

$$UAAA := [-2, -2, 0] \quad (22)$$

> $mult1(UAAA, A)$;

$$0, -2, -2 \quad (23)$$

> $UAAAA := [0, -2, -2]$;

$$UAAAA := [0, -2, -2] \quad (24)$$

> $mult4(UAAAA, A, A, A)$;

$$0, 4, 4 \quad (25)$$

> $UAAAAAAA := [0, 4, 4]$;

$$UAAAAAAA := [0, 4, 4] \quad (26)$$

> $mult4(UAAAAAAA, A)$;

$$-8, 0, 4 \quad (27)$$

> $UA8 := [-8, 0, 4]$;

$$UA8 := [-8, 0, 4] \quad (28)$$

> $mult4(UA8, B, B, E)$;

$$-8, -8, -20 \quad (29)$$

>
u = phi(a + bz + czˆ2) = a + 21b + 21ˆ2c = -8 + 21*-8 + 21ˆ2 * -20 = -8996 v = 2ˆ4*3*5*7 = 1680 vˆ2 mod 9263 = 6448
uˆ2 mod 9263 = 6448
gcd(9263, 1680 + 8996) = 157
gcd(9263, 1680 - 8996) = 59
> $igcd(9263, 10676)$;

$$157 \quad (30)$$

> $igcd(9263, 7316)$;

$$59 \quad (31)$$

> $ifactor(9263)$;

$$(59)\,(157) \quad (32)$$

7. Compare the methods for integer factorisation you have seen in the module and summarises their strengths and weaknesses, including the size of numbers that can be factorised, usability and

whether or not they work for a broad range of numbers.

Answer: Integer factorisation methods we studied in this module include:

(a) Fermat's Factorisation
(b) Euler's Factorisation
(c) Pollard's p-1 method
(d) Pollard's Rho Method
(e) Quadratic Sieve Factorisation and
(f) Number Field Sieve Factorisation

Fermat's Factorisation

Fermat's factorisation is based on the idea that if $x^2 - n$ is a perfect square, let's say $x^2 - n = y^2$, then $n = (x + y)(x - y)$ where x can be found by trials using the numbers $\lfloor\sqrt{(n)}\rfloor + 1, \lfloor\sqrt{(n)}\rfloor + 2, \lfloor\sqrt{(n)}\rfloor + 3, etc.$ This is a general purpose algorithm and efficiently runs for those numbers with one factor close to $sqrt(n)$.

Euler's Factorisation

It uses the principle that if a number can be expressed as sum of squares in two different ways, say $n = a^2 + b^2$ and $n = c^2 + d^2$, then $n = gcd(n, a - c)$. It's easy to factorise n if we know such numbers, but the decomposition into sum of squares is a difficult problem and many numbers do not have such decompositions.

Pollard's p-1 Method

It uses Fermat's Little Theorem. It says if $p|n$, then $p = gcd(a^{p-1} - 1, n)$. If $p - 1$ is B-smooth, then it guarantees finding a factor. It means time efficiency of this algorithm depends on how large B is. Hence it's good for numbers with small prime factors. This is a special purpose algorithm which is being used to prevent for factorisation attacks.

Pollard's Rho Method

Pollard's Rho Method is a probabilistic algorithm and it finds smaller factors faster. In the worst-case, it runs for up to $\sqrt{(n)}$ cycles, which is the same for trial division. Also, it do not guarantee a factor and hence we may need to reinitialise the seed values and restart the searching. This algorithm is not good for applications requiring higher levels of correctness.

Quadratic Sieve Factorisation

Quadratic Sieve algorithm is a general purpose algorithm which guarantees a factor in sub-exponential time complexity for large composite integers with digit size ranging from 40 to 100. It uses the relations $F(T) = T^2 - N$ to do the sieving of B-smooth numbers in the range $\sqrt{N} - M \leq x \leq \sqrt{N} + M$. It becomes a challenge to factorise using Q.S. as choice for B-Smooth numbers are rare for very large numbers. As this algorithm uses Gaussian Elimination to find linearly independent sets, the size of B matters as Gaussian Elimination is $\mathbf{O}(B^3)$. B-factoring M numbers makes the performance bound by the selection of M value as well. An optimum bound for B is approximately [3] $e^{\frac{1}{2}\sqrt{\ln N \ln\ln N}}$

Number Field Sieve

Number Field Sieve is the best general purpose Integer Factorisation algorithm present today. It efficiently factor numbers of digit size 110 or more. The running time for NFS is estimated [3] to be $e^{\frac{64}{9}^{\frac{1}{3}}+O(1))(\ln N)^{\frac{1}{3}}(\ln \ln N)^{\frac{2}{3}}}$. NFS involves an extensive search for a large number of smooth elements and linear algebra computations on large matrices. Hence using NFS will be an overkill for factoring smaller integers.

References

[1] Jeffrey J. Holt, John W. Jones *Discovering Number Theory.*

[2] Matthew E. Briggs *An Introduction to the General Number Field Sieve.*

[3] Elin Margrete Trondsen, *The Number Field Sieve*, Norwegian University of Science and Technology, 2012

[4] Dr Graham Taylor-Russell, *Lecture Notes - Number Theory for Cryptography*, London Metropolitan University

[5] ARJEN K. LENSTRA, *Integer Factoring*