MA7010 – Number Theory for Cryptography - Assignment 2

Ajeesh Thattukunnel Vijayan

January 11th 2024

1 Answers

1. Lower Range = 600, Upper Range = 750. Consider all the numbers n in your range. Divide the set into two subsets: A - the subset consisting of all n where there is at least one primitive root modulo n; B – the subset consisting of all n where no primitive roots exist modulo n

Answer: The below json snippets show the two sets:

```json
{
  "Numbers With Primitve Roots (A)": [
    "601","607","613","614","617","619","622","625",
    "626","631","634","641","643","647","653","659",
    "661","662","673","674","677","683","686","691",
    "694","698","701","706","709","718","719","722",
    "727","729","733","734","739","743","746"
  ]
}

{
  "Numbers Without Primitive Roots (B)": [
    "600","602","603","604","605","606","608","609",
    "610","611","612","615","616","618","620","621",
    "623","624","627","628","629","630","632","633",
    "635","636","637","638","639","640","642","644",
    "645","646","648","649","650","651","652","654",
    "655","656","657","658","660","663","664","665",
    "666","667","668","669","670","671","672","675",
    "676","678","679","680","681","682","684","685",
    "687","688","689","690","692","693","695","696",
    "697","699","700","702","703","704","705","707",
    "708","710","711","712","713","714","715","716",
    "717","720","721","723","724","725","726","728",
    "730","731","732","735","736","737","738","740",
    "741","742","744","745","747","748","749","750"
  ]
}
```

Listing 1: List of Numbers With and Without Primitive Roots

The Rust code for generating the above result is below:

```rust

///
/// Returns a vec of primitive roots for the integer
///
```

```rust
    /// # Arguments
    /// * n: BigInt
    ///
    /// Steps:
    /// This function uses trial and error to find primitive roots
    /// associated to an Integer
    ///
    /// 1. Find all coprime numbers less than `n`
    ///    (coprime_nums_less_than_n)
    /// 2. $\phi(n)$ = total number of coprimes
    /// 3. Find all the divisors of $\phi(n)$. Order of an
    ///    element in the Modulo n group will be equal to
    ///    any of the divisor values.
    /// 4. Find the order of each of the coprimes to n one by one
    ///    (skip 1 from the list of
    ///    coprimes as 1 is a trivial root) (`use utils::modular_pow)
    /// 5. if order of a coprime integer equals $\phi(n)$, that coprime
    ///    is a primitive root
    ///
    /// The above steps are executed aginst all coprimes to n and
    /// returns an integer vector with primitive roots
    ///
    pub fn primitive_roots_trial_n_error(n: &BigInt) -> Vec<BigInt> {
      let mut primitive_roots: Vec<BigInt> = Vec::new();
      let mut has_primitive_roots: bool = false;

      let nums_coprime_n: Vec<BigInt> = coprime_nums_less_than_n(n);
      let phi_n = BigInt::from(nums_coprime_n.len());
      //
      let divisors_phi_n = divisors_of_n(&phi_n);

      for a in nums_coprime_n {
        let mut has_order_phi: bool = true;
        for order in divisors_phi_n.iter() {
          if modular_pow(&a, order, n) == BigInt::one() {
            if *order != phi_n {
              has_order_phi = false;
            }
          }
        }

        if has_order_phi {
          primitive_roots.push(a);
          has_primitive_roots = true;
          break;
        }
      }

      if has_primitive_roots {
        let orders_coprime_phi_n: Vec<BigInt> =
 coprime_nums_less_than_n(&phi_n);
        // first coprime number is 1 and we are skipping that
        // when calculating power
        for order in orders_coprime_phi_n.iter().skip(1) {
          primitive_roots.push(modular_pow(&primitive_roots[0], order,
 n));
        }
      }

      primitive_roots.sort();

      for (i, num) in primitive_roots.clone().iter().enumerate() {
```

```
      if num == &BigInt::one() {
        primitive_roots.remove(i);
        continue;
      }

      if modular_pow(num, &phi_n, n) != BigInt::one() {
        primitive_roots.remove(i);
      }
    }

    primitive_roots
  }

  ///
  /// Generates a list of integers less than n and co-prime to n.
  ///
  pub fn coprime_nums_less_than_n(n: &BigInt) -> Vec<BigInt> {
    let mut coprimes: Vec<BigInt> = Vec::new();
    let r = range(BigInt::from(1u64), n.clone());

    for num in r {
      if n.gcd_euclid(&num) == BigInt::one() {
        coprimes.push(num)
      }
    }
    coprimes.sort();
    coprimes
  }

  ///
  /// Get list of divisors of a number n > 2
  ///
  pub fn divisors_of_n(n: &BigInt) -> Vec<BigInt> {
    let mut divisors: Vec<BigInt> = Vec::new();
    let mut primes = vec![BigInt::from(2u64)];
    let p_factors_n = n.prime_factors(&mut primes);
    let p_factors_n = p_factors_n
    .iter()
    .map(|(p, _)| p.clone())
    .collect::<Vec<BigInt>>();

    for p in p_factors_n {
      let mut i = 0;
      loop {
        let pow = p.pow(i);
        if n % &pow == BigInt::zero() {
          divisors.push(n / &pow);
          divisors.push(pow);
          i += 1;
        } else {
          break;
        }
      }
    }
    divisors.sort();
    divisors.dedup();
    divisors
  }

```

Listing 2: Primitve Roots Calculation

2. a. Explain why we can always find a primitive root modulo p when p is a prime.

Answer: (Not complete)

Theorem 1 (Euler's Theorem) *Suppose that $m \geq 1$ and $(a, m) = 1$, then $a^{\phi(m)} = 1(\mod m)$, where $\phi(m)$ is Euler's Totient function which yields the number of integers less than m and relatively prime to m.*

A special case occurs when m is a prime number, which is called Fermat's Little theorem. When m is a prime, the number of integers less than m and relatively prime to m equal $m - 1$. i.e., $\phi(m) = m - 1$.

b. Express the number of primitive roots that exist modulo p using the Euler Totient function and show that your answer correctly predicts the number of primitive roots for all primes in your given range.

Answer: The number of primitive roots associated with an integer n is given by $\phi(\phi(n))$ When n is a prime, namely p, $\phi(\phi(p)) = \phi(p-1)$. The below table verifies this value against the number calculated using trial and error for all primes in the range $600 \leq p \leq 750$.

Prime	Primitive Roots Count - Trial and Error	$\phi(p-1)$
601	160	160
607	200	200
613	192	192
617	240	240
619	204	204
631	144	144
641	256	256
643	212	212
647	288	288
653	324	324
659	276	276
661	160	160
673	192	192
677	312	312
683	300	300
691	176	176
701	240	240
709	232	232
719	358	358
727	220	220
733	240	240
739	240	240
743	312	312

Table 1: Primitive Roots Count

The Rust code for generating the above result is below. It calls the function listed in code 2.

```

                    PrimitiveRootsCommands::Ass2Question2b(r) => {
            let start = r.start;
            let end = r.end;

            let mut result: Vec<HashMap<String, String>> = Vec::new();
            let (primes_in_range, _) =
    find_primes_in_range_trial_division_parallel(start, end);
            for p in primes_in_range.iter() {
```

```
            let primitive_roots = primitive_roots_trial_n_error(p);
            let phi_phi_n = euler_totient_phi(&(p - BigInt::one()));
            let mut item: HashMap<String, String> = HashMap::new();
            item.insert("Prime".to_string(), p.to_string());
            item.insert("Euler_Totient(p-1)".to_string(), phi_phi_n.
    to_string());
            item.insert(
            "Prim Roots Count - Trial and Error".to_string(),
            primitive_roots.len().to_string(),
            );
            result.push(item);
          }
          println!("{}", serde_json::to_string_pretty(&result).unwrap
    ())
        }

```

Listing 3: Primitive Roots - Euler's Totient Function Verification

We can use the below command to see the above result:

```

.\target\release\nt-assignments.exe primitive-roots ass2-question2b -s 600 -e 750

```

Listing 4: Verify Primitive Roots Counting using Totient Function

c. For the same range as Question 1 use the command ifactors in Maple to find the set C whose elements consist of numbers of the form $p^k(p > 2, k \geq 1)$ or $2p^k(p > 2, k \geq 1)$

Number	Form	Number	Form
601	601^1	674	$2^1 \times 337^1$
607	607^1	677	677^1
613	613^1	683	683^1
614	$2^1 \times 307^1$	686	$2^1 \times 7^3$
617	617^1	691	691^1
619	619^1	694	$2^1 \times 347^1$
622	$2^1 \times 311^1$	698	$2^1 \times 349^1$
625	5^4	701	701^1
626	$2^1 \times 313^1$	706	$2^1 \times 353^1$
631	631^1	709	709^1
634	$2^1 \times 317^1$	718	$2^1 \times 359^1$
641	641^1	719	719^1
643	643^1	722	$2^1 \times 19^2$
647	647^1	727	727^1
653	653^1	729	3^6
659	659^1	733	733^1
661	661^1	734	$2^1 \times 367^1$
662	$2^1 \times 331^1$	739	739^1
673	673^1	743	743^1
746	$2^1 \times 373^1$	-	-

Table 2: Numbers of the form p^k, $2p^k$

d. Hence form a conjecture about when primitive roots do and don't exist

3. Suppose n has the form $n = pq$ where p and q are different primes both > 2.

(a) What is $\phi(n)$ in terms of p and q?

Answer: $\phi(n) = \phi(p.q) = \phi(p).\phi(q) = (p-1).(q-1)$

(b) Suppose a is relatively prime to pq. Explain why

i. $a^{p-1} \equiv 1 \mod p$

Answer: Given p and q are distinct primes. Since $(a, pq) = 1$, a is relatively prime to both p and q. Hence by Fermat's Little Theorem, $a^{p-1} \equiv 1 mod p$.

ii. $a^{q-1} \equiv 1 \mod q$

Answer: Given p and q are distinct primes. Since $(a, pq) = 1$, a is relatively prime to both p and q. Hence by Fermat's Little Theorem, $a^{q-1} \equiv 1 mod q$.

iii. $m = lcm(p-1, q-1)$ is less than $(p-1)(q-1)$

Answer: Since both p and q are odd primes, $p-1$ and $q-1$ are even. Let $p-1 = 2j$ and $q-1 = 2k$. Then $(p-1, q-1) = (2j, 2k) = 2(j, k)$. We can see that there will be a factor of 2 at a minimum when number are even. LCM is given by $lcm(p-1, q-1) = \frac{(p-1)(q-1)}{gcd(p-1,q-1)}$, which means $m = lcm(p-1, q-1)$ equals $(p-1)(q-1)$ only when $gcd(p-1, q-1) = 1$. But here we have $gcd > 1$ and hence $m = lcm(p-1, q-1) < (p-1)(q-1)$

iv. $a^m \equiv 1 \mod (p-1)(q-1)$

Answer:

(c) Hence explain why numbers of the form n have no primitive roots.

Answer: Suppose $n = p.q$, where p and q are primes has primitive roots. This means there exists $a \in (\mathbb{Z}/pq\mathbb{Z})^{\times}$ such that $ord_{pq}(a)$ will be $m = \phi(n) = \phi(p.q) = (p-1).(q-1)$. Also $gcd(a, pq) = 1$.

Also,

$$a^m \equiv 1 \pmod{pq} \tag{1}$$

$$\iff a^m \equiv 1 \pmod p, a^m \equiv 1 \pmod q \text{(By Chinese Remainder Theorem)} \tag{2}$$

$$\iff m \equiv 0 \pmod{p-1}, m \equiv 0 \pmod{q-1} \tag{3}$$

(Because by Fermatś Little Theorem, $a^{p-1} \equiv 1 \pmod p$ and $a^{q-1} \equiv 1 \pmod q$)

$$\iff (p-1)|m, (q-1)|m \tag{4}$$

$$\iff lcm(p-1, q-1)|m \tag{5}$$

This means that $ord_p(a) = lcm(p-1, q-1) < (p-1)(q-1)$ as we have seen in 3(b)iii and it's a contradiction from our initial assumption that $n = p.q$ has primitive roots.

(d) Show that all numbers of the form $n = pq$ (p and q both odd primes) in your range are included in set B.

Answer:


```
          {
            "Numbers Without Primitive Roots (B)": [
            "600","602","603","604","605","606","608","609",
            "610","611","612","615","616","618","620","621",
            "623","624","627","628","629","630","632","633",
            "635","636","637","638","639","640","642","644",
            "645","646","648","649","650","651","652","654",
            "655","656","657","658","660","663","664","665",
            "666","667","668","669","670","671","672","675",
            "676","678","679","680","681","682","684","685",
            "687","688","689","690","692","693","695","696",
            "697","699","700","702","703","704","705","707",
            "708","710","711","712","713","714","715","716",
```

```
                "717","720","721","723","724","725","726","728",
                "730","731","732","735","736","737","738","740",
                "741","742","744","745","747","748","749","750"
                ]
            }

```

Listing 5: List of Numbers Without Primitive Roots

And the below table shows the list of numbers of th form $p.q$ in the range $600 \leq n \leq 750$, which is a subset of the set B above 5.

Number	Form	Number	Form
611	$13^1 \times 47^1$	687	$3^1 \times 229^1$
623	$7^1 \times 89^1$	689	$13^1 \times 53^1$
629	$17^1 \times 37^1$	695	$5^1 \times 139^1$
633	$3^1 \times 211^1$	697	$17^1 \times 41^1$
635	$5^1 \times 127^1$	699	$3^1 \times 233^1$
649	$11^1 \times 59^1$	703	$19^1 \times 37^1$
655	$5^1 \times 131^1$	707	$7^1 \times 101^1$
667	$23^1 \times 29^1$	713	$23^1 \times 31^1$
669	$3^1 \times 223^1$	717	$3^1 \times 239^1$
671	$11^1 \times 61^1$	721	$7^1 \times 103^1$
679	$7^1 \times 97^1$	723	$3^1 \times 241^1$
681	$3^1 \times 227^1$	731	$17^1 \times 43^1$
685	$5^1 \times 137^1$	737	$11^1 \times 67^1$
-	-	745	$5^1 \times 149^1$
-	-	749	$7^1 \times 107^1$

Table 3: Numbers of the form $p.q$

4. Use the BabyStepsGiantSteps algorithm to find discrete logarithms x of b mod n for the primitive root a for each of the two examples assigned to you in the table below. Verify that your answer is correct by calculating $a^x \mod m$ by hand using the method of modular exponentiation.

Note: Somehow I couldn't make it work the Baby Steps Giant Steps Algorithm as we learned in the class. I checked the Wikipedia and it's the same as in the class. I do not know where did it go wrong. I was getting a smaller value that expected. So I followed the steps from some Youtube videos(Video1, Video2). The steps are similar with some minor variations in the values we calculate. Hope that's fine.

(a) ***Answer:*** Given $a = 21, b = 47, n = 71$. We want to solve for t in the congruence: $21^t \equiv 47 \pmod{71}$

We have $\phi(71) = 70$.

Step 1. Set m = $\lceil\sqrt{71}\rceil = 9$
Step 2. Calculating $a^{mj} \pmod{71}; 0 \leq j < m$

j	$a^{mj} \pmod{71}$	j	$a^{mj} \pmod{71}$	j	$a^{mj} \pmod{71}$
0	$21^{9.0} = 1$	3	$21^{9.3} = 35$	6	$21^{9.6} = 18$
1	$21^{9.1} = 42$	4	$21^{9.4} = 50$	7	$21^{9.7} = 46$
2	$21^{9.2} = 60$	5	$21^{9.5} = 41$	8	$21^{9.8} = 15$

Step 3. Solve for $b.a^{-i}; 0 \leq i < m$

i	$b.a^{-i} \pmod{71}$	i	$b.a^{-i} \pmod{71}$
0	$47.21^0 = 47$	4	$47.21^{-4} = 47.21^{66} = 69$
1	$47.21^{-1} = 47.21^{69} = 9$	5	$47.21^{-5} = 47.21^{65} = 54$
2	$47.21^{-2} = 47.21^{68} = 41$	6	$47.21^{-6} = 47.21^{64} = 33$
3	$47.21^{-3} = 47.21^{67} = 29$	7	$47.21^{-7} = 47.21^{63} = 32$
-	-	8	$47.21^{-8} = 47.21^{62} = 59$

Step 4. We found a collision in both the tables for $value = 41$ where $i = 2$ and $mj = 9 \times 5$

Step 5. We calculate $t = (mj + i) \pmod{71} = (9 \times 5 + 2) \pmod{71} \equiv 47 \pmod{71}$
$\implies 21^{47} \equiv 47 \pmod{71}$

Step 6. Verifying the answer using Fast Modular Exponentiation:

$$21^2 \equiv 15 \pmod{71}$$
$$\therefore 21^4 = (21^2)^2 = 15^2 \pmod{71} \equiv 12 \pmod{71}$$
$$\implies 21^8 = (21^4)^2 = 12^2 \equiv 2 \pmod{71}$$
$$\implies 21^{16} = (21^8)^2 = 2^2 \equiv 4 \pmod{71}$$
$$\implies 21^{32} = (21^{16})^2 = 4^2 \equiv 16 \pmod{71}$$
$$\implies 21^{40} = 21^{32} \times 21^8 \pmod{71} \equiv 16 \times 2 \pmod{71} \equiv 32 \pmod{71}$$

We will calculate now $21^7 \pmod{71}$using the below steps:

The binary representation for $7 = [111] \sim [d_2 d_1 d_0]$

Let $a = 1$ and $s = 21$

$k = 0$: Since $d_k = 1, a = a \times s = 21 \pmod{71}, s = s^2 = 15 \pmod{71}$

$k = 1$: Since $d_k = 1, a = a \times s = 31 \pmod{71}, s = s^2 = 12 \pmod{71}$

$k = 2$: Since $d_k = 1, a = a \times s = 17 \pmod{71}$,

$\implies 21^7 \equiv 17 \pmod{71}$

$\therefore 21^{47} = 21^{40} \times 21^7 = 32 \times 17 \equiv 47 \pmod{71}$ and hence the answer

(b) ***Answer:*** Given $a = 26, b = 24, n = 53$. We want to solve for t in the congruence:
$26^t \equiv 24 \pmod{53}$

Step 1. Set m $= \lceil\sqrt{53}\rceil = 8$

Step 2. –

Step 3. Calculating $a^{mj} \pmod{53}; 0 \leq j < m$ & Solve for $b.a^{-i} \pmod{53}; 0 \leq i < m$ (Step 2 and 3 tables below side-by-side)

$26^{-1} \equiv 27 \pmod{53}$

j	a^{mj} (mod 71)
0	$26^{8.0} \equiv 1$
1	$26^{8.1} \equiv 47$
2	$26^{8.2} \equiv 36$
3	$26^{8.3} \equiv 49$
4	$26^{8.4} \equiv 24$
5	$26^{8.5} \equiv 15$
6	$26^{8.6} \equiv 16$
7	$26^{8.7} \equiv 10$

Table 4: Step 2

i	$b.a^{-i}$ (mod 53)
0	$24.26^{0} = 24.26^{52} = 26$
1	$24.26^{-1} = 24.26^{51} = 5$
2	$24.26^{-2} = 24.26^{50} = 43$
3	$24.26^{-3} = 24.26^{49} = 20$
4	$24.26^{-4} = 24.26^{48} = 13$
5	$24.26^{-5} = 24.26^{47} = 27$
6	$24.26^{-6} = 24.26^{46} = 52$
7	$24.26^{-7} = 24.26^{45} = 2$
8	$24.26^{-8} = 24.26^{44} = 49$

Table 5: Step 3

Step 4. We found a collision in both the tables for $value = 49$ where $i = 8$ and $mj = 8 \times 3$

Step 5. We calculate $t = (mj + i) \pmod{53} = (8 \times 3 + 8) \pmod{53} \equiv 32 \pmod{53}$
$\implies 26^{32} \equiv 24 \pmod{53}$

Step 6. Verifying the answer using Fast Modular Exponentiation:

$$26^2 \equiv 40 \pmod{53}$$
$$\therefore 26^4 = (26^2)^2 = 40^2 \equiv 10 \pmod{53}$$
$$\implies 26^{16} = (26^4)^4 = 10^4 \equiv 36 \pmod{53}$$
$$\implies 26^{32} = (26^{16})^2 = 36^2 \equiv 24 \pmod{53} \text{ and hence the answer}$$

5. Use the Pohlig Helmann algorithm to find in the cyclic group of order n with the generating element a for both the examples assigned to you below. Verify your answer in Maple.

(a) ***Answer:*** $a = x^{11}, b = x^{41}, n = 343$

$n = 343 = 7^3$
We will write G as $G = \{x^i | 0 \leq i \leq 342, x^{342} = 1\}$. Also x^{11} generates G as $(11, 343) = 1$.
So we set $p = 7, e = 3, g = x^{11}, h = x^{41}$

Step 1. Setting $x_0 = 0$ and let $n = p^e, p^{e-1} = p^2 = 49$
When $k = 0$:
$s = g^{p^{e-1}} = g^{n/p} = (x^{11})^{49} = x^{539} = x^{196}$
$h_0 = (g^{-x_0} \times h)^{p^{e-1}} = h^{49} = (x^{41})^{49} = x^{294}$
Since $p = 7$, test for $d_0 \in \{0, 1, 2, 3, 4, 5, 6\}$ satisfying $s^{d_0} = h_0$
$\therefore$, for $d_0 = 5$, we have $s^{d_0} = h_0$, so $d_0 = 5$
$x_1 = x_0 + p^0.d_0 = 0 + 1.5 = 5$

Step 2. When $k = 1$ we have $x_1 = 5, p^{e-2} = p^1 = 7$
$h_1 = (g^{-x_1} \times h)^{p^{e-2}} = (g^{-5} \times h)^7 = (x^{-55} \times x^{41})^7 = x^{-98} = x^{245}$
Searching for $d_1 \in \{0, 1, 2, 3, 4, 5, 6\}$ satisfying $s^{d_1} = h_1$
$\therefore, r = 3$ satisfies the condition. So $d_1 = 3$
$x_2 = x_1 + p^1.d_1 = 5 + 7 \times 3 = 26$

Step 3. When $k = 2$, we have $x_2 = 26, p^{e-3} = 1$
$h_2 = (g^{-x_2} \times h)p^{e-3} = (g^{-26} \times h)^1 = x^{-286} \times x^{41} = x^{-245} = x^{98}$
Searching for d_2, we get $d_2 = 4$
$x_3 = x_2 + p^2.d_2 = 26 + 49 \times 4 = 222$

$x = 222$ is the logarithm we wanted.

Below is the Maple Verification result:

$G = \{x^i | 0 \leq i \leq 342, x^{342} = 1\}$
$\gcd(11, 343) = 1, x^{11} generates the Group.$
$p := 7; e := 3; g := x^{11}; h := x^{41};$

$$p := 7$$
$$e := 3$$
$$g := x^{11}$$
$$h := x^{41} \tag{6}$$

Step1:
$x0 := 0;$

$$x0 := 0 \tag{7}$$

$s := x^{11 \cdot 49 \mod 343}$; $h0 := x^{41 \cdot 49 \mod 343}$;

$$s := x^{196}$$
$$h0 := x^{294} \tag{8}$$

$Searching \boldsymbol{for} d0; d0 = 5 satisfies s^{d0} = h0$
$d0 := 5;$

$$d0 := 5 \tag{9}$$

$x^{196 \cdot 5 \mod 343}$;

$$x^{294} \tag{10}$$

$x1 := x0 + p^0 \cdot d0;$

$$x1 := 5 \tag{11}$$

Step 2:
$h1 := x^{(-55+41) \cdot 7 \mod 343}$;

$$h1 := x^{245} \tag{12}$$

$Searching \boldsymbol{for} d1; d1 = 3 satisfies s^{d1} = h1$
$d1 := 3;$

$$d1 := 3 \tag{13}$$

$x^{196 \cdot 3 \mod 343}$;

$$x^{245} \tag{14}$$

$x2 := x1 + p^1 \cdot d1;$

$$x2 := 26 \tag{15}$$

Step 3:

$h2 := x^{-286+41 \mod 343}$;

$$h2 := x^{98} \tag{16}$$

$Searching\,\boldsymbol{for}\,d2; d2 = 4 satisfiess^{d2} = h2$
$d2 := 4;$

$$d2 := 4 \tag{17}$$

$x^{196\cdot 4 \mod 343}$;

$$x^{98} \tag{18}$$

$x3 := x2 + p^2 \cdot d2;$

$$x3 := 222 \tag{19}$$

$x3\, is\, our\, logarithm$

(b) ***Answer:*** $a = x^{13}, b = x^{157}, n = 3267$

$n = 3267 = 3^3 \times 11^2 = 27 \times 121$
We will write G as $G = \{x^i | 0 \leq i \leq 3266, x^{342} = 1\}$. Also x^{13} generates G as $(13, 3267) = 1$.
Step 1.

$$g1 = g^{121} = x^{13\times 121 \pmod{27}} = x^7$$
$$h1 = h^{121} = x^{157\times 121 \pmod{27}} = x^{16}$$

We will need to find the logarithm of $h1 = x^{16}$ in the cyclic group of order 27 generated by $g1 = x^7$. With some trial and error, we get $logx^{16} = 10$, i.e., $x^{7 \pmod{27}} = x^{16}$. Hence we get the below congruence:

$$x \equiv 10 \pmod{27} \tag{20}$$

Step 2.

$$g2 = g^{27} = x^{13\times 27 \pmod{121}} = x^{109}$$
$$h2 = h^{27} = x^{157\times 27 \pmod{121}} = x^4$$

Let's find the logarithm of $h2 = x^4$ in the cyclic group of order 121 generated by $g2 = x^{109}$. With some trial and error, we get $logx^4 = 40$, i.e., $x^{109\times 40 \pmod{121}} = 4$. We get the following congruence:

$$x \equiv 40 \pmod{121} \tag{21}$$

Step 3. We will now need to solve the congruences 20 and 21. We will employ Chinese Remainder Theorem for that. Suppose we have a system of congruences as below:

$$\begin{cases} x & \equiv b_1 \pmod{n_1} \\ x & \equiv b_2 \pmod{n_2} \\ x & \equiv b_3 \pmod{n_3} \\ & . \\ & . \\ & . \\ x & \equiv b_k \pmod{n_k} \end{cases} \tag{22}$$

CRT states that the above congruence has a unique modulo $N = n_1.n_2.n_3...n_k$ solution if each n_i are pairwise coprime and is given by:

$$x = \sum_{i=1}^{k} b_i e_i (N/n_i)$$
$$\text{where } e_i = (N/n_i)^{-1} \pmod{n_i}$$

Restating our equations below:

$$\begin{cases} x & \equiv 10 \pmod{27} \\ x & \equiv 40 \pmod{121} \end{cases} \tag{23}$$

$$n_1 = 27, n_2 = 121$$
$$N = n = 27 = 3267$$
$$b_1 = 10, b_2 = 40$$
$$e_1 = 121^{-1} \pmod{27} = 25$$
$$e_2 = 27^{-1} \pmod{121} = 9$$
$$\therefore x = 10 \times 25 \times 121 + 40 \times 9 \times 27 = 766 \pmod{3267}$$

$x = 766$ is the logarithm we wanted.

Below is the Maple Verification result:

$\mathit{find\_log} := \boldsymbol{proc}(n, a, m)$
$\boldsymbol{description}$ "Find log of a";
$\boldsymbol{for} i \boldsymbol{from} 1 \boldsymbol{to} n$
$\boldsymbol{do}$
$\boldsymbol{if} a \cdot i \quad \boldsymbol{mod}\ n = m$
$\boldsymbol{then}$
$\boldsymbol{return} i;$
$\boldsymbol{fi};$
$\boldsymbol{enddo};$
$\boldsymbol{endproc};$

```
>

```

$\mathit{ifactor}(3267)$

$$(3)^3 (11)^2 \tag{24}$$

$n := 3^3 \cdot 11^2;$

$$n := 3267 \tag{25}$$

g = xˆ13, h = xˆ157, n = 3267; ¡g¿ generatees the group of order 3267;
Steps 1:
$g1 := x^{13 \cdot 11^2 \mod 27};$

$$g1 := x^7 \tag{26}$$

$h1 := x^{157 \cdot 11^2 \mod 27};$

$$\mathit{h1} := x^{16} \tag{27}$$

We need to find the log of h1 = x^16 in the cyclic group of order 27 generated by g1 = x^7. By trial and error, we get log(h1) = 10

$\mathit{find\_log}(27, 7, 16);$

```
>

```

$$10 \tag{28}$$

So our first congruence is x1 = 10 mod 27 — (1)

Step 2:

$\mathit{g2} := x^{13 \cdot 3^3 \mod 121};$

$$\mathit{g2} := x^{109} \tag{29}$$

$\mathit{h2} := x^{157 \cdot 3^3 \mod 121};$

$$\mathit{h2} := x^{4} \tag{30}$$

We need to find the log of h2 = x^4 in the cyclic group of order 121 generated by g2 = x^109; using the proc find_log above, it is = 118
$\mathit{find\_log}(121, 109, 4);$

$$40 \tag{31}$$

We get our second congruence as: x2 = 40 mod 121 — (2)

Hence we need to find the unique solution to x = 10 mod 27, and x = 40 mod 121 using Chinese Remainder Theorem.

$\mathit{with}(\mathit{NumberTheory});$

$$[\mathit{AreCoprime}, \mathit{CalkinWilfSequence}, \mathit{CarmichaelLambda}, \mathit{ChineseRemainder}, \mathit{ContinuedFraction}, \mathit{ContinuedFractionPolynomial}, \mathit{CyclotomicPolynomial}, \mathit{Divisors}, \mathit{FactorNormEuclidean}, \mathit{HomogeneousDiophantine}, \mathit{ImaginaryUnit}, \mathit{InhomogeneousDiophantine}, \mathit{IntegralBasis}, \mathit{InverseTotient}, \mathit{IsCyclotomicPolynomial}, \mathit{IsMersenne}, \mathit{IsSquareFree}, \mathit{IthFermat}, \mathit{IthMersenne}, \mathit{JacobiSymbol}, \mathit{JordanTotient}, \mathit{KroneckerSymbol}, \mathit{Landau}, \mathit{LargestNthPower}, \mathit{LegendreSymbol}, \mathit{M\ddot{o}bius}, \mathit{ModExtendedGCD}, \mathit{ModularLog}, \mathit{ModularRoot}, \mathit{ModularSquareRoot}, \mathit{Moebius}, \mathit{MultiplicativeOrder}, \mathit{M\ddot{o}bius}, \mathit{NearestLatticePoint}, \mathit{NextSafePrime}, \mathit{NumberOfIrreduciblePolynomials}, \mathit{NumberOfPrimeFactors}, \Omega, \Phi, \mathit{PrimeCounting}, \mathit{PrimeFactors}, \mathit{PrimitiveRoot}, \mathit{PseudoPrimitiveRoot}, \mathit{QuadraticResidue}, \mathit{Radical}, \mathit{RepeatingDecimal}, \mathit{RootsOfUnity}, \mathit{SimplestRational}, \mathit{SumOfDivisors}, \mathit{SumOfSquares}, \mathit{ThueSolve}, \mathit{Totient}, \lambda, \mu, \phi, \mathrm{pi}, \sigma, \tau, \varphi] \tag{32}$$

$\mathit{ChineseRemainder}([10, 40], [27, 121]);$

$$766 \tag{33}$$

6. Use the Pollard Rho method to verify your answer to the first example you were allocated in Question 4.

Name	b	n	a	Method
Ajeesh	47	71	21	BabyStepGiantStep
Ajeesh	24	53	26	BabyStepGiantStep
Ajeesh	x^{41}	343	x^{11}	Pohlig Hellmen
Ajeesh	x^{157}	3267	x^{13}	Pohlig Hellmen

Table 6: Table Listing the Problems Allocation to Individuals

Answer: The below table shows the execution of the Pollard Rho algorithm to generate the data table:

Pollard Rho Execution Data						
i	x1	a1	b1	x2	a2	b2
1	21	1	0	15	2	0
2	15	2	0	2	8	0
3	12	4	0	16	8	2
4	2	8	0	27	10	2
5	23	8	1	44	21	4
6	16	8	2	10	42	10
7	52	9	2	1	43	11
8	27	10	2	15	18	22
9	19	20	4	2	2	18
10	44	21	4	16	2	20
11	9	21	5	27	4	20
12	10	42	10	44	9	40
13	68	43	10	10	18	12
14	1	43	11	1	19	13

Table 7: Pollard Rho Execution Data

From the table, we can see that $x1 = x2$ at the 14^{th} iteration. The corresponding a1, a2, b1, b2 values are: $i = 14$, $a1 = 43$, $a2 = 19$, $b1 = 11$, $b2 = 13$.

Let t be the logarithm of b. Calculation of the logarithm is below:

$$(b2 - b1).t \equiv (a1 - a2) \pmod{p-1}$$
$$\implies (13 - 11).t \equiv (43 - 19) \pmod{70}$$
$$\implies 2.t \equiv 24 \pmod{70} \tag{34}$$

Let $d = gcd(2, 70) = 2$

Divide Eqn. (34) by d, we get: $t \equiv 12 \pmod{35}$

Since gcd = 2, there are two solution to Eqn. (35) and the solutions are:

$$\{12, 12 + \frac{70}{2}\} = \{12, 47\} \tag{35}$$

When $t = 47$ our congruence $21^t \equiv 47 \pmod{71}$ is satisfied and hence the logarithm of 47 is 47. Thus we have verified solution of the problem in 4a.

References

[1] C R Jordan & D A Jordan *MODULAR MATHEMATICS Groups* .

[2] Dr. Ben Fairbairn *GROUP THEORY Solutions to Exercises.*

[3] *https://github.com/Ssophoclis/AKS-algorithm/blob/master/AKS.py*