Homework 3

Graded

Student

Kevin Vuong

View or edit group

Total Points

24.75 / 25 pts

Question 1

Question #1

3 / 3 pts

✔ **– 0 pts** Correct

– 0.25 pts Incorrect unique prime factorization for 1b

– 0.5 pts Did not include unique prime factorization for 1b

– 1 pt See comment

Question 2

Question #2

4 / 4 pts

✔ **– 0 pts** Correct

– 0.25 pts No check in python for #2a

– 0.5 pts #2b incorrect

1 should be positive 147 -- work is correct

Question 3

Question #3

2 / 2 pts

✔ **– 0 pts** Correct

– 0.25 pts gave negative values for #3b

Question 4

Question #4

3 / 3 pts

✔ **– 0 pts** Correct

Question 5

Question #5

3 / 3 pts

✔ **– 0 pts** Correct

– 0.25 pts did not make positive

Question 6

Question #6

4 / 4 pts

✔ **– 0 pts** Correct

– 0.25 pts Incorrect generators for Z7

– 0.25 pts Incorrect generators for Z9

– 0.5 pts 6b incorrect

Question 7

Question #7

2 / 2 pts

✔ **– 0 pts** Correct

– 0.25 pts #7 incorrect

Question 8

Question #8

1.75 / 2 pts

– 0 pts Correct

✔ **– 0.25 pts** #8 incorrect

14328 is correct final answer

Question 9

Question #9

2 / 2 pts

✔ **– 0 pts** Correct

Questions assigned to the following page: 1 and 2

Homework 3

Instructions: Solutions must be typed using LaTeX. All work and the link to any accompanying Python code should be uploaded to Gradescope by 11:59PM on Friday, February 9, 2024.

1. (3 pts) Write (or find) Python code that determines if a number is prime. Use your code to state whether each of the following is prime or composite. If the number is composite, write its unique prime factorization.

   (a) $7,349$ (b) $127,483,237$

   (a) $7,349$ is a prime. (PYTHON CODE HERE)

   (b) $127,483,237$ is not a prime. It's prime factorization is $7 * 281 * 648411$. (PYTHON CODE HERE)

2. (4 pts)

   (a) Use the Euclidean Algorithm to find $\gcd(42823, 6409)$ by hand. Check your answer in Python.

   (b) Use the Extended Euclidean Algorithm to find u, v such that $\gcd(42823, 6409) = u \cdot 42823 + v \cdot 6409$.

   (a)

$$\begin{aligned} 42823 &= 6409 * 6 + 4369 \\ 6409 &= 4369 * 1 + 2040 \\ 4369 &= 2040 * 2 + 289 \\ 2040 &= 289 * 7 + 17 \\ 289 &= 17 * 17 + 0 \end{aligned}$$

   We look at the second to last row and see that the remainder is 17. Thus, $\gcd(42823, 6409) = 17$. This is Python Verified. (PYTHON CODE HERE)

   (b)

   we first solve for the remainders.

$$\begin{aligned} 4369 &= 42823 - 6409 * 6 \\ 2040 &= 6409 - 4369 \\ 289 &= 4369 - 2040 * 2 \\ 17 &= 2040 - 289 * 7 \end{aligned}$$

   Then compute

$$\begin{aligned} 17 &= 2040 - 289 * 7 \\ &= 6409 - 4369 - (4369 - 2040 * 2) * 7 \\ &= 6409 - 4369 - 7 * 4369 + 14 * 2040 \\ &= 6409 - 8 * 4369 + 14(6409 - 4369) \\ &= 15 * 6409 - 22 * 4369 \\ &= 15 * 6409 - 22(42823 - 6409 * 6) \\ &= 147 * 6409 - 22 * 42823 \end{aligned}$$

   Thus, our values u, v such that $\gcd(42823, 6409) = u * 42823 + v * 6409$ are $u = -22, v = 147$

Questions assigned to the following page: 3, 5, and 4

3. (2 pts)
   (a) Give two negative values of x such that $x \equiv 2 (\mod 7)$.
   (b) Give two positive values of x such that $x \equiv 3 (\mod 5)$.

   (a) $x = -5, x = -12$
   (b) $x = 8, x = 13$

4. (3 pts) Reduce each of the following.

   (a) $72 (\mod 60)$ (b) $-4 (\mod 6)$ (c) $-19 (\mod 12)$

   (a)
   $$72 (\mod 60) \equiv 12 (\mod 60)$$
   (b)
   $$-4 (\mod 6) \equiv 2 (\mod 6)$$
   (c)
   $$-19 (\mod 12) \equiv 5 (\mod 5)$$

5. (3 pts) Find $13^{-1} (\mod 35)$ by hand using the Euclidean Algorithm. Check your answer in Python.
   We first factor $\gcd(35, 13)$. Via Euclidean Algorithm.

   $$\gcd(35, 13)$$

   $$\begin{aligned} 35 &= 13 * 2 + 9 \\ 13 &= 9 * 1 + 4 \\ 9 &= 4 * 2 + 1 \\ 4 &= 4 * 1 + 0 \end{aligned}$$

   We look at the 2nd to last equation and see that the remainder is 1. Thus $\gcd(35, 13) = 1$.
   We now factor 1 into its Bezout Identity. We first solve for the remainder.

   $$\begin{aligned} 9 &= 35 - 13 * 2 \\ 4 &= 13 - 9 * 1 \\ 1 &= 9 - 4 * 2 \end{aligned}$$

   Then,

   $$\begin{aligned} 1 &= 9 - 2 * 4 \\ &= 9 - 2(13 - 9) \\ &= 9 - 2 * 13 + 2 * 9 \\ &= 3 * 9 - 2 * 13 \\ &= 3 * (35 - 2 * 13) - 2 * 13 \\ &= 3 * 35 - 8 * 13 \end{aligned}$$

   Thus, $1 = 3 * 35 - 8 * 13$. We now apply (mod 35).

   $$1 (\mod 35) \equiv 3 * 35 - 8 * 13 (\mod 35)$$

Questions assigned to the following page: 5, 6, 7, 8, and 9

$$\implies 1(\text{mod } 35) \equiv -8 * 13(\text{mod } 35)$$
$$\implies 13^{-1}(\text{mod } 35) \equiv -8(\text{mod } 35)$$
$$\implies 13^{-1}(\text{mod } 35) \equiv 27(\text{mod } 35).$$

Thus,

$$13^{-1}(\text{mod } 35) \equiv 27(\text{mod } 35)$$

This answer was verified in Python. (PYTHON CODE HERE)

6. (4 pts) Assume addition for the following.
   (a) List the generators of Z_7 and Z_9.
   (b) In general, what are the generators of Z_n?

   (a) The generators of Z_7 are $1, 2, 3, 4, 5, 6$. Furthermore the generators of Z_9 are $1, 2, 4, 5, 7, 8$

   (b) In general, we define the generators of Z_n to be all numbers co-prime to n.

7. (2 pts) Suppose you choose $p = 19$ and $q = 17$ for your private key with the RSA Cryptosystem. Using $e = 283$ as your encryption exponent, calculate d.

   We first calculate n and ϕ.

$$n = pq = 323$$
$$\phi = (p-1)(q-1) = 288$$

   we want to solve $d \equiv e^{-1}(\text{mod } 288)$ (PYTHON CODE HERE)

$$d = 115 \equiv e^{-1}(\text{mod } 288)$$

8. (2 pts) Suppose you know that Alice and Bob are using the RSA Cryptosystem with public key $(n, e) = (179329, 28473)$. You intercept Bob's ciphertext message to Alice, $c = 120159$. Decrypt it and recover his plaintext message.

   We factorize n with pollard's p-1 algorithm. We get

$$p = 461, q = 389.$$
$$\implies \phi = 178480$$

   Then, using the Euclidean Algorithm

$$ed \equiv 1(\text{mod } 178480)$$
$$\implies d = 44147(\text{mod } 178480)$$
$$\implies m \equiv 120159^{44147}(\text{mod } 178480)$$

9. (2 pts) Explain two ways that an attacker could universally break RSA.

   (1). One way an attacker could universally break RSA is if they could find a way to efficiently factor n into p and q in a reasonable amount of time. (Thus, she'd be able to derive the private key on her own). This could likely be done with a Quantum Computer, but not a classical one (Since integer factorization is presumed to be $NP - Hard$)

   (2). Another way an attacker could break RSA is if they could come up with their own way to undo the encryption without the decryption algorithm. i.e., the attacker would find an easy way to compete f^{-1}, if f is the one-way/encryption function of RSA.