Module 10.5: Euler's Totient Function and Modular Exponentiation

Gregory V. Bard

December 11, 2020

- This module is designed to teach you about Euler's Totient function, which is usually just called "Phi," and modular exponentiation. The skills learned here will be crucial when learning about RSA.
- There is a with-answers version, and a without-answers version.
- In the with-answers version of this workbook, the black ink represents the question, and the blue ink represents the answer. The red ink is additional information that would normally only be calculated with technology (e.g. the computer-algebra system called Sage, or perhaps Maple).

A Brain Teaser, Before We Get Started

Among other things, in this module you will learn how to answer some questions that were very popular in math competitions when I was in high school (in the 1990s). As you go through this module, you might want to think about how you would approach these problems. Examples include

- What is the last digit of 1993^{1992}?
- What are the last two digits of 2001^{1963}?
- What are the last two digits of 2003^{1975}?
- What are the last two digits of 1929^{1963}?
- What are the last three digits of 2003^{1604}?
- What are the last three digits of 1453^{2001}?
- ... but keep in mind that we did these problems without a calculator. I've designed these problems to be done with a very simple calculator, such as a 4-function or 5-function calculator.

Part One: Euler's Totient Function, $\Phi(N)$

One of the key results of Module 10-2: Modular Inverses, is that we have a quick and easy test to determine, for any fixed integers b and N, whether b is invertible mod N or not. We know that b is invertible mod N if and only if b is coprime to N. We should also remember that the words "b is coprime to N" formally mean that $\gcd(b, N) = 1$. Informally, it means that the prime factorizations of b and of N have no primes in common.

This enables us to ask a very straightforward question, which can be stated in three mathematically equivalent ways:

- Of the integers z such that $0 < z < N$, how many are coprime to N?
- Of the integers z such that $0 < z < N$, how many have $\gcd(z, N) = 1$?
- In the world of the integers mod N, how many numbers have inverses?

Clearly, we can define a function where N is the input, and the output is the answer to that one question above that we wrote in three mathematically equivalent, but verbally different ways. The great mathematician Leonhard Euler (1707–1783) found many uses of this function, and it is called $\Phi(N)$.

The symbol Φ is the Greek letter Phi, and it is their letter for the "F" sound. In mathematics, it is pronounced like the second syllable of the word "defy," or like the word "fly" without the "L" sound. Actual Greeks pronounce it like the word "fee."

In summary, when we say that $\Phi(a) = b$, what we mean is that

- There are b integers z such that $0 < z < a$ and $\gcd(z, a) = 1$.
- There are b integers z such that $0 < z < a$ and z is invertible mod a.
- There are b integers z such that $0 < z < a$ and z is coprime to a.
- There are b integers z such that $0 < z < a$ and z is relatively prime to a.

Keep in mind that each of those four bullets above is saying precisely the same thing.

Question 10-5-1

In this question, you will derive the formula for $\Phi(p)$, for any prime integer p.

- Let p be a prime integer. Regardless of the choice of prime p, which numbers in the world of the integers mod p are invertible?
- Based on that fact alone, how many numbers in the world of the integers mod p are invertible?
- This now reveals a formula for $\Phi(p)$ that will work for any prime p. What is that formula?

Computing $\Phi(N)$ when $N = pq$

One way that we can compute $\Phi(N)$ for any N is by setting up a `for` loop, iterating through all the integers z with $0 < z < N$, and computing the gcd. We just need to count the number of times that we get $\gcd(z, N) = 1$.

The real reason that we're learning about Φ is because of its role in the RSA cryptosystem. The modulus in RSA is always the product of two distinct huge primes. I should point out that a prime which is 100 digits long is definitely too small to be used with RSA in the real world. One of many consequences of that fact is that there is no hope of using a `for` loop to compute $\Phi(N)$. We simply must find another way.

Luckily, there are two easy roads to a quick and simple formula! Our first approach will involve the following theorem:

> For any two positive integers N_1 and N_2, with N_1 coprime to N_2,
>
> $$\Phi(N_1 N_2) = \Phi(N_1)\Phi(N_2)$$

However, proving the theorem above requires The Chinese Remainder Theorem or "CRT." Proving the CRT requires an entire lecture of its own and it would be inconvenient for us to have such a detour at this time. While the theorem above answers our question, since we haven't proven it, we are left unsatisfied. Nonetheless, if we accept this theorem above, we obtain that if p and q are distinct prime integers then

$$\Phi(pq) = \Phi(p)\Phi(q) = (p-1)(q-1)$$

which is a formula that we will use very often when studying RSA.

Unlike theology, in mathematics, we should never take anything on faith. We should always require proof. Therefore, I'm very happy to present you with a combinatorial proof of this formula, especially because the proof is very compact and easy to understand. In fact, we'll do the proof as an exercise.

Question 10-5-2

Before we do the combinatorial proof that $\Phi(pq) = (p-1)(q-1)$, we need the following tool. When we consider the integers $\{1, 2, 3, \ldots, kp\}$, how many of them are divisible by p?

Question 10-5-3

While we normally think of the integers mod N as being $\{0, 1, 2, \ldots, N-1\}$, we can also think of them as $\{1, 2, 3, \ldots, N\}$, because $0 \equiv N \bmod N$. We can say that zero and N are two names for the same number in the world of the integers mod N.

Looking at the integers $\{1, 2, 3, \ldots, N\}$, we will denote the subsets that are divisible by p with the symbol $\mathcal{P}$, and the subsets that are divisible by q with the symbol $\mathcal{Q}$.

- What is the size of the set $\mathcal{P}$? Hint: you could reread the previous problem if you have forgotten it.

- What is the size of the set $\mathcal{Q}$?
- What is the size of the set $\mathcal{P} \cap \mathcal{Q}$?
- What is the size of the set $\mathcal{P} \cup \mathcal{Q}$? Hint: use the inclusion-exclusion principle.
- When $N = pq$, the product of two distinct primes, we know that the only primes in the prime factorization of N are p and q. Therefore, the only way that any z could possibly have $\gcd(z, N) > 1$ is if z is divisible by p or by q. With this in mind, how many integers in $\{1, 2, 3, \ldots, N\}$ have $\gcd(z, N) > 1$?
- How many integers in $\{1, 2, 3, \ldots, N\}$ have $\gcd(z, N) = 1$?

Question 10-5-4

- What is $\Phi(187)$?
- What is $\Phi(19)$?
- What is $\Phi(35)$?
- What is $\Phi(101)$?

Question 10-5-5

- A reliable person tells you that $\Phi(5491) = 4896$. How many integers z with $0 < z < 5491$ are non-invertible mod 5491? How many are invertible mod 5491?
- A reliable person tells you that $\Phi(125) = 100$. How many integers z with $0 < z < 125$ are non-invertible mod 125? How many are invertible mod 125?

Question 10-5-6

- Of the numbers that exist in the world of the integers mod 25, which are not invertible?
- What is $\Phi(25)$?
- Of the numbers that exist in the world of the integers mod 4, which are invertible? which are not invertible?
- What is $\Phi(4)$?
- Can we compute $\Phi(100)$ without writing out all 100 numbers that exist in the world of the integers mod 100?

Question 10-5-7

- What is $\Phi(10)$?
- Surely $10 \times 10 = 100$. Thus we can write

$$\Phi(10) \times \Phi(10) = 4 \times 4 = 16$$

however, this is a problem because we have already computed that

$$\Phi(100) = 40 \neq 16$$

Is this a contradiction? Is mathematics broken? What has gone wrong here?

Question 10-5-8

Suppose that I travel to the world of the integers mod N, and that I select a number from that world, chosen uniformly at random. What is the probability that the number has no inverse mod N?

- If $N = 4$, then what is the probability?
- If $N = 19$, then what is the probability?
- If $N = 35$, then what is the probability?
- If $N = 100$, then what is the probability?

Part Two: Exploring Modular Exponentiation

We're going to use Sage via the SageMathCell interface for this exploration. Alternatively, you can use CoCalc.com or even a local installation of Sage, but that's overkill.

There is a link to the SageMathCell code that I've written for the modular arithmetic exploration, on the same webpage where you found this module. In case that link is somehow missing or broken, the code is also given on the last page of this module, so that you can cut-and-paste it.

When you click on the link, the default example is executed. These are the powers of 4 mod 31. Specifically, we are computing

$$4^0, 4^1, 4^2, 4^3, 4^4, 4^5, \ldots \mod 31$$

and we obtain

1, 4, 16, 2, 8, 1, 4, 16, 2, 8, 1, 4, 16, 2, 8, 1, 4, 16, 2, 8, 1, 4, 16, 2, 8, 1, 4, 16, 2, 8, ...

where the colors have been added in this PDF file only to help you see the pattern. We could mathematically describe this pattern as follows:

- if $k \equiv 0 \bmod 5$, then $4^k \equiv 1 \bmod 31$.
- if $k \equiv 1 \bmod 5$, then $4^k \equiv 4 \bmod 31$.
- if $k \equiv 2 \bmod 5$, then $4^k \equiv 16 \bmod 31$.
- if $k \equiv 3 \bmod 5$, then $4^k \equiv 2 \bmod 31$.
- if $k \equiv 4 \bmod 5$, then $4^k \equiv 8 \bmod 31$.

Next, just change `base = 4` into `base = 27`, but don't change anything else. This will tell SageMathCell to compute

$$27^0, 27^1, 27^2, 27^3, 27^4, 27^5, \ldots \bmod 31$$

when you click the large gray "Evaluate" button. We obtain

1, 27, 16, 29, 8, 30, 4, 15, 2, 23, 1, 27, 16, 29, 8, 30, 4, 15, 2, 23, 1, 27, 16, 29, 8, 30, 4, 15, 2, 23, 1, 27, 16, 29, 8, 30, 4, 15, 2, 23, 1, 27, 16, 29, 8, 30, 4, 15, 2, 23, ...

We could mathematically describe this pattern as follows:

- if $k \equiv 0 \bmod 10$, then $27^k \equiv 1 \bmod 31$.
- if $k \equiv 1 \bmod 10$, then $27^k \equiv 27 \bmod 31$.
- if $k \equiv 2 \bmod 10$, then $27^k \equiv 16 \bmod 31$.
- if $k \equiv 3 \bmod 10$, then $27^k \equiv 29 \bmod 31$.
- if $k \equiv 4 \bmod 10$, then $27^k \equiv 8 \bmod 31$.
- if $k \equiv 5 \bmod 10$, then $27^k \equiv 30 \bmod 31$.
- if $k \equiv 6 \bmod 10$, then $27^k \equiv 4 \bmod 31$.
- if $k \equiv 7 \bmod 10$, then $27^k \equiv 15 \bmod 31$.
- if $k \equiv 8 \bmod 10$, then $27^k \equiv 2 \bmod 31$.
- if $k \equiv 9 \bmod 10$, then $27^k \equiv 23 \bmod 31$.

Question 10-5-9

Take care to keep the modulus at 31. Use the explorer tool in SageMathCell to answer the following questions, by changing the value of `base` and clicking Evaluate.

- What is the sequence of powers of 26?
- Using the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, describe the powers of 26 mod 31.

- What is the sequence of powers of 8?
- Using the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, describe the powers of 8 mod 31.
- What is the sequence of powers of 5?
- Using the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, describe the powers of 5 mod 31.
- What is the sequence of powers of 30?
- Using the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, describe the powers of 30 mod 31.

What's the moral of the story?

When we compute the start of the infinite sequence $b^0, b^1, b^2, b^3, b^4, \ldots$ mod N, we see that it repeats one specific finite length sequence, over and over. We can capture that repetition using modular arithmetic. However, patterns are extremely non-obvious.

Of course, since there are only N numbers in the world of the integers mod N, the infinite sequence sequence $b^0, b^1, b^2, b^3, b^4, \ldots$ mod N must contain repeats. Even if we only continued the sequence for $N + 1$ terms, we'd have to repeat, because the integers mod N does not contain $N + 1$ distinct numbers. What is neat is that the pattern repeats in such a precise and orderly way.

Part Three: The Order of an Element

Naturally, we should ask ourselves why the pattern repeats in such a precise and orderly way. We need the following theorem:

Suppose for some positive integer f, $g^f \equiv 1 \bmod N$. Then for any $k = qf + r$, where q and r are integers and $0 \leq r < f$, it will be the case that $g^k \equiv g^r \bmod N$. The proof is only one long line:

$$g^k = g^{qf+r} = g^{qf} g^r = (g^f)^q g^r \equiv 1^q g^r = 1 g^r = g^r \bmod N$$

Now if we let k iterate through the positive integers (or non-negative integers), then $r \equiv k \bmod f$. That's why the sequence proceeds in such an orderly fashion. However, we could describe the infinite sequence

$$1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, 1, 30, \ldots$$

as an infinite repetition of the 6-symbol subsequence (1, 30, 1, 30, 1, 30) or of the 4-symbol subsequence (1, 30, 1, 30). In order to describe it well, we should choose the shortest possible subsequence. That will come from choosing the smallest possible value for f.

The smallest positive integer k such that $g^k \equiv 1 \bmod N$ is called "the order of g in the integers mod N." Sometimes mathematicians get sloppy and call this "the order of g mod N." Rarely, one will see the abbreviation $\text{ord}_N(g) = k$, but it is more common to write $\text{ord}(g) = k$ when the choice of modulus (for some specific problem) has been made clear.

In addition to k being the smallest possible positive integer exponent such that $g^k \equiv 1 \bmod N$, the value of k gives us the shortest possible subsequence that gets repeated to generate the infinite sequence $g^0, g^1, g^2, g^3, g^4, \ldots$. Moreover, if asked for g^n for some specific integer n, we can reduce $n \bmod k$ to be able to figure out the actual value of g^n.

Question 10-5-10

Looking back over the exploration of modular exponentiation mod 31, answer the following:

- What is the order of 4 mod 31?
- What is the order of 27 mod 31?
- What is the order of 26 mod 31?
- What is the order of 8 mod 31?
- What is the order of 5 mod 31?
- What is the order of 30 mod 31?
- Did we encounter any members of the integers mod 31 whose order is not a divisor of $\Phi(31) = 30$?

Question 10-5-11

Just for fun, let's compute something in the complex numbers, $\mathbb{C}$. Let $i = \sqrt{-1}$. Using your pencil or pen, and not a computer or calculator, answer the following:

- What is the sequence of powers of i?
- Using the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, describe the powers of $i \in \mathbb{C}$.
- What is the sequence of powers of 2?
- Can we use the notation that I used before to describe the powers of 4 mod 31 in terms of mod 5, and the powers of 27 mod 31 in terms of mod 10, to describe the powers of $2 \in \mathbb{C}$?

- What is the order of i in $\mathbb{C}$?

We write $\text{ord}(2) = \infty$ to indicate that the sequence of powers of 2 will never repeat. Our definition of order, for modular arithmetic, required the order to be a positive integer. So if we want to go beyond modular arithmetic, we need to broaden our definition.

We can say that $\text{ord}(g)$ is the size of the set of numbers which can be written as g^z for some positive integer z. There are only four complex numbers that can be written as positive integer powers of i, but there are infinitely many complex numbers that can be written as powers of 2.

The Etymology of "Totient": Long ago, when these theories were being developed, the elements in (equivalently, the members of) a mathematical object such that

$$g^0, g^1, g^2, g^3, g^4, g^5, \ldots$$

repeated a finite-length sequence over and over again were called the "torsion elements" of that object. The other elements or members were called the "non-torsion elements." Like $2 \in \mathbb{C}$, the non-torsion elements have a sequence that goes on forever, without ever repeating even one entry. This was seen as a very fundamental distinction among the elements/members of a mathematical object.

In modular arithmetic, every member of the integers mod N is a torsion element, and there are no "non-torsion elements." The totient function is meant to measure this torsion, just as "weight" measures how much something "weighs." Well anyway, we needed a word and Euler gave us one.

Part Pi: The Euler Totient Theorem

Euler's Totient theorem is the following:

For any positive integers b and N, b is coprime to N if and only if $b^{\Phi(N)} \equiv 1 \bmod N$.

Amazingly, we now have four ways of saying the same thing:

- b is invertible mod N.
- $\gcd(b, N) = 1$.
- $b^{\Phi(N)} \equiv 1 \bmod N$.
- b is coprime to N.
- (Some textbooks rewrite the last one as "b is relatively prime to N.")

I will not present a proof of Euler's Totient theorem at this time, as it would lead us off topic.

However, we can behave like scientists working in the physical sciences, and we can conduct some experiments. There should be an applet powered by SageMathCell, on the same

webpage where you found this module, with the title "Exploring Euler's Totient Theorem." Click on that. You can see a census of the world of the integers mod 72. For each number in that world, we ask if the gcd with the modulus equals 1 (that's P) and then we ask if that number raised to the power of $\Phi(N)$ is congruent to 1 mod N (that's Q). You can scroll down and see that either P and Q are both true, or alternatively, P and Q are both false, for every number in the world of the integers mod 72. This comprises a numerical experiment verifying Euler's Totient Theorem for the special case of $N = 72$ only.

Now you can change the modulus to something else. I recommend trying $N = 31$, which is prime, and $N = 35$, which is a product of two primes.

Part Four: The Upstairs-Downstairs Principle

Here is a corollary of Euler's Totient Theorem. For any positive integers b and N, if b is coprime to N, and $u_1 \equiv u_2 \bmod \Phi(N)$, then $b^{u_1} \equiv b^{u_2} \bmod N$.

Given Euler's Totient theorem, the proof of the above corollary is only two steps. Let $\varphi = \Phi(N)$, and suppose $u_1 \equiv u_2 \bmod \varphi$. This means there is some integer z such that $u_1 = u_2 + z\varphi$. Then,

$$b^{u_1} = b^{u_2+z\varphi} = b^{u_2}b^{z\varphi} = b^{u_2}(b^{\varphi})^z \equiv b^{u_2}1^z = b^{u_2}1 = b^{u_2} \bmod N$$

Thus $b^{u_1} \equiv b^{u_2} \bmod N$ as desired.

Finally, the upstairs-downstairs principle states that if d_1 is coprime to N, $d_2 \equiv d_1 \bmod N$, and $u_1 \equiv u_2 \bmod \Phi(N)$, then $(d_1)^{u_1} \equiv (d_2)^{u_2} \bmod N$.

This is only a minor modification of the corollary, because of the simple fact that when working mod N, if $d_1 \equiv d_2 \bmod N$ then I can substitute d_1 for d_2, or vice-versa, at any moment.

Therefore, when asked for $d_1^{u_1} \bmod N$, we will reduce the base d_1 ("the downstairs") mod N to get d_2, and the exponent u_1 ("the upstairs") mod $\Phi(N)$ to get u_2. The upstairs-downstairs principle guarantees that

$$(d_1)^{u_1} \equiv (d_2)^{u_2} \bmod N$$

Furthermore, because d_1 might be much smaller than d_2, and because u_1 might be much smaller than u_2, this will give us a much simpler form that hopefully is more easily computable.

By the way, while this technique has been known since the days of Euler, the name "upstairs-downstairs principle" is my own nickname for it.

Question 10-5-12

Use the upstairs-downstairs principle to answer the following questions. You should be able to do them with a very simple calculator, such as a 4-function or 5-function calculator.

- What is $5393^{763} \bmod 55$?

- What is 1849^{1066} mod 77?
- What is 1921^{1914} mod 101? (Hint: $2^{14} = 16,384$.)

Question 10-5-13

Now you can solve the 1990s era high-school math-competition problems that I mentioned at the start of the module. By the way, it is useful for us to know that $\Phi(100) = 40$ and $\Phi(1000) = 400$. You'll compute that yourself in Question 10-5-17.

Of course, asking "what are the last two digits of junk" is the same thing as asking "what is junk mod 100." Therefore, we can use the upstairs-downstairs principle to solve these problems. Of course, if we are only asked for the last digit, we use mod 10, and if we are asked for the last three digits, we use mod 1000.

- What is the last digit of 1993^{1992}?
- What are the last two digits of 2001^{1963}?
- What are the last two digits of 2003^{1975}?
- What are the last two digits of 1929^{1963}?
- What are the last three digits of 2003^{1604}?
- What are the last three digits of 1453^{2001}?

Part Five: Computing Euler Phi in General

Thus far, we only have formulas for computing $\Phi(N)$ when N is either prime, or alternatively, the product of two primes.

Question 10-5-14

In Question 10-5-1, we saw that when we consider the integers $\{1, 2, 3, \ldots, kp\}$, that k of them are divisible by p. We will now use that information to compute

- Now consider the integers $\{1, 2, 3, \ldots, p^n\}$, where p is some prime integer, and n some positive integer. How many integers in that set are divisible by p?
- For any positive integer z, when we compute $\gcd(z, p^n)$, why is the only possible answer a power of p? Note: since $p^0 = 1$, we consider 1 to be a power of p.
- Consider the world of the integers mod p^n. Why is it true that a number in that world is invertible if and only if it is not a multiple of p?
- In the world of the integers mod p^n, how many numbers are not invertible?

- In the world of the integers mod p^n, how many numbers are invertible?
- For any prime p and any positive integer n, what is a quick and easy formula for $\Phi(p^n)$?

Question 10-5-15

Compute the Euler Totient function $\Phi(N)$ for the following numbers.

- What is $\Phi(16)$?
- What is $\Phi(256)$?
- What is $\Phi(8)$?
- What is $\Phi(64)$?

Question 10-5-16

Now use the results of the previous question to answer the following:

- In hexadecimal, what is the last digit of 7^{8881}?
- In hexadecimal, what are the last two digits of $9^{12,803}$?
- In octal, what is the last digit of 3^{1997}?
- In octal, what are the last two digits of $5^{32,003}$?

Question 10-5-17

Compute the Euler Totient function $\Phi(N)$ for the following numbers.

- What is $\Phi(72)$?
- What is $\Phi(120)$?
- What is $\Phi(360)$?
- What is $\Phi(1000)$?

Here is the code for the modular exponentiation explorer.

```
base = 4
modulus = 31 # must be less than 150

# this code will explore the sequence
# base^0, base^1, base^2, base^3, base^4, ...
# in modular arithmetic.
###############
# don't change anything below unless you
# know what you are doing!

stop = 4*modulus

# the next four lines ensure that both
# base and modulus are positive integers
assert base in ZZ
assert modulus in ZZ
assert base > 0
assert modulus > 0

# this next line will prevent the program
# from running if the modulus is 150 or bigger,
# so that valuable computing time is not wasted.
assert modulus < 150

seen_before = [ ]
# seen_before is a list, initially empty

for k in range(0, stop+1):
    answer = pow( base, k, modulus )

    print(base, "^", k, "=", answer, "mod", modulus, end="")

    if answer in seen_before:
        # if the answer is soemthing we've seen before, say so
        print(" (This has been seen before.)")
    else:
        # if the answer hasn't been seen before, well, we've
        # seen it now, so put it into the list seen_before
        print()
        seen_before.append( answer )
```

Here is the code for the Euler's Totient Theorem explorer.

```
modulus = 72

###################
################### Don't change anything below here...
################### ... unless you know what you're doing

assert modulus in ZZ
assert modulus > 0
assert modulus < 250

phi = euler_phi( modulus )

print("P represents: k is invertible mod", modulus)
print("Q represents: k ^", phi, "= 1 mod", modulus)
print()

for k in range(0, modulus):
    the_gcd = gcd(k, modulus)

    print("gcd(", k, ",", modulus, ") =", end=" ")
    print(the_gcd, "so P is", end=" ")
    if (the_gcd==1):
        print("true.")
    else:
        print("false.")


    answer = pow(k, phi, modulus)

    print(k, "^", phi, "=", answer, "mod", end=" ")
    print(modulus, "so Q is", end=" ")

    if (answer==1):
        print("true.")
    else:
        print("false.")

    print()
```