Lesson Fourteen

Math 6080 (for the Masters Teaching Program), Summer 2020

14. Euler's ϕ Function. We've tested numerically in Lesson Twelve that for any fixed modulus m, the primes distribute themselves evenly among the remainders:

$$p\%m = r$$

that are relatively prime to m (i.e. $\gcd(m, r) = 1$). The number of such remainders (between 1 and $m - 1$) is the output of the **Euler ϕ function**:

$$\phi(m)$$

Let's start by writing Python code to compute this function with brute force:

(1) Write a function def gcd(x,y) that returns the gcd of x and y.

(2) Initiate a counter phi = 0

(3) For r in range($1.m - 1$), call the function gcd to get gcd(m,r). If this is 1, then increase the counter phi by one.

(4) print the counter phi.

Notice that when $m = p$ is a **prime** number:

$$\phi(p) = p - 1$$

because each gcd(p,r) is a divisor of the prime p (and less than p), so it must be 1.

Similarly, when $m = p^2$ is the square of a prime, then only the remainders that are **multiples** of p fail to be relatively prime to p^2. Between 1 and p^2, there are $p - 1$ of these:

$$p, 2p, ..., (p-1)p, \text{ so}$$
$$\phi(p^2) = (p^2 - 1) - (p - 1) = p^2 - p$$

(the number of numbers from 1 to p^2 minus the number of multiples of p). Similarly,

$$\phi(p^n) = (p^n - 1) - (p^{n-1} - 1) = p^n - p^{n-1}$$

is the number of numbers from 1 to p^n minus the number of multiples of p.

So what about the numbers that are **not** primes or powers of primes? (like 6)

Chinese Remainder I. Let x and y be natural numbers and consider the function:

$$f(r) = (r\%x, r\%y)$$

that maps emainders for the modulus xy to ordered pairs of remainders for the moduli x and y. The function is a map:

$$f : \{0, 1, ..., xy - 1\} \to \{0, 1, ..., x\} \times \{0, 1, ..., y\}$$

between two sets of xy elements.

Theorem. If x and y are relatively prime, then f is a bijective map.

Proof. Using the enhanced Euclid's algorithm, we can solve:

$$ax + by = 1$$

with integers a and b because $\gcd(x, y) = 1$. Now suppose that

$$(s, t) \in \{0, 1, ..., x\} \times \{0, 1, ..., y\}$$

Then

$$g(s,t) = (ax)t + (by)s \ \% \ xy \in \{0, ..., xy-1\}$$

satisfies:

(a) $g(s,t)\%x = (by)s\%x(bx)x + (by)s\%x = s\%x$ and

(b) $g(s,t)\%y = (ax)t\%y = (ax)t + (ay)t\%y = t\%y.$

In other words, $g(s,t)$ is the **inverse function** of $f(r)$. So $f(r)$ is bijective.

Example. Take $x = 5$ and $y = 7$. Then running the enhanced Euclid gives:

$$(3)5 + (-2)7 = 1$$

so the function $g(s,t)$ is:

$$g(s,t) = 15t - 14s$$

Lets' try it out.

$$g(3,5)\%35 = 15(5) - 14(3) = 75 - 42 = 33 \text{ and } f(33) = (3,5). \text{ Check!}$$

Exercise. Implement this inverse function with a Python program, prompting the user for x and y and two remainders s and t, and outputting the value $g(s,t)$.

This is a good party trick. Ask a friend to give your the remainder of their age when it is divided by 11 and 13, and then find the age of the friend.

Corollary. If x and y are relatively prime, then:

$$\phi(xy) = \phi(x)\phi(y)$$

Proof. The bijective function f maps numbers relatively prime to xy to ordered pairs of numbers relatively prime to x and to y, respectively. □

Strategy for Computing the Euler ϕ function of n.

Step 1. Factor n as a product of powers of primes (this is tough when n is big!).

Step 2. Use the formulas for $\phi(p^n)$ and the Chinese Remainder Theorem I

Examples. (i) $\phi(45) = \phi(5 \cdot 3^2) = \phi(5)\phi(3^2) = 5(3^2 - 3) = 30.$

(i) $\phi(144) = \phi(2^4 \cdot 3^2) = (2^4 - 2^3)(3^2 - 3) = 8 \cdot 6 = 48.$

(Check these against your program.)

Exercise. Write a function def factor(n) to factor a number n, returning an ordered list of the prime factors. Then call this function from a program that uses it to compute the value of the phi function for n. Try this out with a large number. It will run much faster than your original program (why?).