SRQR: Euler's Theorem (April 13, 2021)

> I'm confused by the proof for the Chinese Remainder Theorem; how does one go from $x \equiv a_i \pmod{n_i}$ to $x = a_1 N_1^{\phi(n_1)} + a_2 N_2^{\phi(n_2)} + \cdots + a_r N_r^{\phi(n_r)}$?

I'm interpretting this question to mean "where does this formula comes from?" rather than "why does this formula work?" To motivate this formula, let's compare the expression

$$x = a_1 N_1^{\phi(n_1)} + a_2 N_2^{\phi(n_2)} + \cdots + a_r N_r^{\phi(n_r)} \tag{0.1}$$

to the expression

$$x = a_1 N_1 x_1 + a_2 N_2 x_2 + \cdots a_r N_r x_r \tag{0.2}$$

that shows up in the proof of the Chinese Remainder Theorem in theorem 4.8.

Notice the difference is just that, where formula (0.1) has $N_i^{\phi(n_i)}$, formula (0.2) instead has $N_i x_i$. We know that $\gcd(N_i, n_i) = 1$. In formula (0.1), we have $N_i^{\phi(n_i)}$, which is congruent to 1 mod n_i by Euler's theorem. In formula (0.2), x_i is chosen to be an inverse of N_i mod n_i, so we have $N_i x_i$ is also congruent to 1 mod n_i.

Stated in yet another way, Euler's theorem tells us that $N_i^{\phi(n_i)-1}$ is an inverse of N_i mod n_i. So we can choose $x_i = N_i^{\phi(n_i)}$ in (0.2) in the second formula and we arrive at formula (0.1).

> In example 7.2 where "the method of successive squaring" takes place, how does it go from $3^4 \equiv 81 \pmod{100}$ to $3^8 \equiv 61 \pmod{100}$? I know that we can calculate it by hand, but is there a quicker way?

Not really. You do have to square 81, which I don't know a quick way of doing. That being said, you might note that since you're reducing mod 100, you don't really need to actually calculate all of the digits of the square. You just need to calculate the last two digits, which is pretty quick to do:

$$\begin{array}{r} 81 \\ \times\, 81 \\ \hline 81 \\ ??8 \\ \hline ??61 \end{array}$$

> We have a corollary to Euler's theorem that is Fermat's theorem. Is it possible to go the other way?

In fact, it is possible! If you take a look at the "Second Proof of Euler's Theorem'' on page 139, you'll see that this is in fact a proof of Euler's theorem using Fermat's theorem! It's a

nice inductive proof. It's not as easy as deriving Fermat's theorem from Euler's, but then again, Euler's theorem is a stronger statement so that should come as no surprise.

> How do we know that the hint for problem 4 is true? I'm not sure where I'm supposed to remember it from. It was: $a^{\phi(n)} - 1 = (a - l)(a^{\phi(n)-1} + \ldots + a^2 + a + 1)$.

The more general fact that you may have seen at some point is that

$$a^k - b^k = (a - b)(a^{k-1} + a^{k-2}b + a^{k-3}b^2 + \cdots + ab^{k-2} + b^{k-1})$$

for any integer k and any a and b. This fact is not very hard to prove: just multiply out the right-hand side! Notice, if we do this, we get

$$(a^k + a^{k-1}b + \cdots + a^2b^{k-2} + ab^{k-1}) - (a^{k-1}b + \cdots + ab^{k-1} + b^k)$$

and all of the terms here cancel, except for a^k and b^k.

> Would you mind talking about CC number 4?

Since $\gcd(a, n) = 1$, we know by Euler's theorem that $a^{\phi(n)} \equiv 1 \pmod{n}$, or, in other words, that $a^{\phi(n)} - 1 \equiv 0 \pmod{n}$. Using the factorization in the hint (and discusseed above), we have

$$(a - 1)(a^{\phi(n)-1} + \cdots + a^2 + a + 1) = a^{\phi(n)} - 1 \equiv 0 \pmod{n}.$$

Since $\gcd(a - 1, n) = 1$, we know that $a - 1$ has an inverse mod n (corollary to theorem 4.7). In other words, there exists an integer b such that $b(a - 1) \equiv 1 \pmod{n}$. So if we take the equation above and multiply through by b, we get

$$a^{\phi(n)-1} + \cdots + a + 1 \equiv b(a - 1)(a^{\phi(n)-1} + \cdots + a + 1) \equiv b \cdot 0 = 0 \pmod{n},$$

which is what we wanted to show.

> I was a bit confused about 7.3.9.

> It seems that problems of this chapter are mostly around very large numbers (like problem 7 and 9). Since what I did — which is just trying all kinds of combinations of factorization randomly until one make sense (for example, I spent 20min just to find a way to write 10000 in terms of combination of 60 and 77) — had cost me a lot of time, I was wondering if there is any way to factorize and combine large numbers like that more easily?

Here's how I would do problem 9 systematically. Notice that

$$\phi(77) = \phi(7 \cdot 11) = 77\left(1 - \frac{1}{7}\right)\left(1 - \frac{1}{11}\right) = 77 \cdot \frac{6}{7} \cdot \frac{10}{11} = 60$$

and that $\gcd(2, 77) = 1$. Euler's theorem thus says that $2^{60} \equiv 1 \pmod{77}$. If I divide 10000 by 60, I get

$$10000 = 166 \cdot 60 + 40$$

which means that

$$2^{10000} = 2^{166 \cdot 60 + 40} = (2^{60})^{144} \cdot 2^{40} \equiv 2^{40} \pmod{77}.$$

So I just have to calculate 2^{40}, and this I can do by binary exponentiation without too much work.

$$\begin{aligned} 2^2 &= 4 \\ 2^4 &= 16 \\ 2^8 &= 256 \equiv 25 \pmod{77} \\ 2^{16} &\equiv 625 \equiv 2 \pmod{77} \\ 2^{32} &\equiv 4 \pmod{77} \end{aligned}$$

Putting the above together, I find that

$$2^{10000} \equiv 2^{40} = 2^{32} \cdot 2^8 \equiv 2 \cdot 25 = 50 \pmod{77}.$$