SRQR: Finding Primitive Roots (April 15, 2021)

> I'm probably overthinking this, but when it says "p is an odd prime," does that just mean $p \neq 2$, or does it mean something like it has to be an odd place in the order of primes (like the 3rd prime, 5th prime, etc)?

It just means $p \neq 2$. I don't know why this sort of obtuse convention for referring to all primes except 2 developed, but it's pretty common.

> Why exactly are primitive roots so significant? Why is 1 mod n so important/interesting?

> For me chapter 8 is taking a lot of digestion time because it seems so unfamiliar — are there applications of primitive roots and totients that could help combat the unfamiliarity?

> Were primitive roots invented (defined?) as a means towards proving some specific thing, or was it just in the in hopes that it would lead somewhere interesting? Do they have extrinsic significance?

My knowledge of mathematical history is woefully weak, so I cannot say why they were originally defined. But I can say that this idea has had a number of applications down the road. In fact, it has both theoretical applications (as in, it's very useful for proving other theorems in math!) and also practical applications. Let me sketch one direction that the practical applications take.

Suppose r is an integer whose order mod n is k. Let's first say that I give you some integer h between 1 and k and ask you to compute $r^h \bmod n$. You can do this very quickly using binary exponentiation. More explicitly, you can do this in logarithmic time (the binary exponentiation algorithm runs in $O(\log h) \leqslant O(\log k)$ time).

On the other hand, let's say I give you an integer a, and I tell you that it can be written as a power of r mod n, but I don't tell you what that power is. In other words, I tell you that there exists *some* integer h between 1 and $\phi(n)$ such that $r^h \equiv a \bmod n$, but I don't tell you what h is. Finding this integer h computationally is called the *Diffie-Hellman problem*. It turns out we have no good algorithm thus far for solving the Diffie-Hellman problem (ie, for explicitly finding this integer h). More explicitly, the only algorithms we have basically involving testing all the integers from 1 up through k, so this algorithm runs roughly in $O(k)$ time.

Now notice that the difference between $\log k$ and k gets bigger as k gets bigger. That means that the asymmetry betweeen computing powers of r, and solving the Diffie-Hellman

problem is as big as possible when k is as big as possible — and that happens when r is a primitive root!

Having as large an asymmetry as possible is extremely useful in cryptography. Many important cryptographic protocols rely on their security for having an asymmetry between these two computations that's as large as possible. The idea is basically that, let's say the world knows r and n. You pick out a secret number h, and you can calculate $a = r^h \bmod n$ very quickly. Now you can reveal the number a to the world, but, if all the relevant numbers are big enough, no one out there will have a computer that's powerful enough to compute h. The fact that you have this "secret" number h allows you to exchange messages securely.

Why is the statement in theorem 8.7 important enough to be its own theorem?

Part of what's happening in section 8.3 is that Burton is proving exactly what numbers have primitive roots. A part of characterizing exactly which numbers have primitive roots is saying which numbers *don't* have primitive roots. Theorem 8.7 tells you that *most* powers of 2 (ie, all of them except 2 and 4) don't have primitive roots. Theorems 8.8 and its corollary continue this thought and characterize other numbers that don't have primitive roots. Then theorem 8.9 and its corollary characterize which numbers *do* have primitive roots.

Theorem 8.10 summarizes all of this information, saying *exactly* which numbers have primitive roots.

How do you prove 2.b?

Could you go over how to do #2b from Section 8.3?

Would you mind going over CC 2(b) please?

Here are two (related) ways of arguing this, without using the book's hint. Suppose r is a primitive root of p^n. We want to show that it's a primitive root mod p.

First method: Suppose s is a primitive root of p. Since $1 \leqslant s \leqslant p-1$, we know that $p \nmid s$, so $\gcd(s, p^n) = 1$. By theorem 8.4, there exists an integer h such that $r^h \equiv s \bmod p^n$, which implies that $r^h \equiv s \bmod p$.

Let k be the order of r mod p. Theorem 8.3 tells us that

$$p - 1 = \text{order of } s \bmod p = \frac{k}{\gcd(h, k)}.$$

Rearranging this tells us that $p - 1 \mid k$. But theorem 8.1 tells us that $k \mid p - 1$, so we must have $k = p - 1$. In other words, k is a primitive root mod p.

Second method: Suppose we have an arbitrary integer a between 1 and $p-1$ (inclusive). Then $\gcd(a, p^n) = 1$, so there exists an integer h such that $r^h \equiv a \bmod p^n$. This implies that $r^h \equiv a \bmod p$. Thus, every integer a between 1 and $p-1$ can be written as a power of $r \bmod p$, so r must be a primitive root of p.