Math 75 notes, Lecture 25

P. Pollack and C. Pomerance

What about the integers?

In the last lecture we discussed some algorithmic problems for polynomials over finite fields. We saw that it is easy (i.e., there is a 'good algorithm') to decide whether or not a polynomial irreducible and to factor it into irreducibles if it isn't. What about the analogous problems for integers? How easy is it to decide whether a number is prime, and if it's not, to break it up into its prime factors?

There is an obvious naive algorithm; trial divide by all $2 \le d \le \sqrt{n}$. This is just as terrible as the analogous algorithm for polynomials discussed in the last lecture, and for the same reason: if n actually is prime, this requires about $\sqrt{n}$ divisions, and even though each of these divisions is fairly quick, their number is exponential in the length of the input (i.e., in $\log n$).

So, just as before, getting to a good algorithm will require some cleverness. Once again, it will be the theory of finite fields that comes to our aid.

Recognizing primes

According to Fermat's little theorem, if n is prime and a is any integer coprime to n, then we have the congruence

$$a^{n-1} \equiv 1 \pmod{n}. \tag{1}$$

This is one of the basic facts about the finite field $\mathbb{F}_n$ that we have made repeated use of in this course. In particular, if n is an odd prime, then $2^{n-1} \equiv 1 \pmod{n}$.

As an example, take $n = 91$. We can easily compute (by the repeated squaring algorithm) that $2^{90} \equiv 64 \pmod{91}$. So 91 cannot be a prime! (In fact, as you may have already noticed, $91 = 7 \cdot 13$.) What is notable about the proof via Fermat's little theorem is that it shows 91 composite without factoring it. Moreover, this Fermat's-little-theorem test is quick: for a general number n, the repeated squaring algorithm allows us to compute $2^{n-1} \pmod{n}$ in about $O(\log n)$ steps. And each step is also fairly quick: we have only to multiply two numbers $< n$ and then reduce the answer modulo n, and both can be done with only $O(\log^2 n)$ basic steps. So given an odd number, it's easy to test if $2^{n-1} \equiv 1 \pmod{n}$, and if it's not, we have a proof that n is composite.

This is all good as far as it goes, but it doesn't go far enough. Take, say, $n = 341$. Then we have $2^{340} \equiv 1 \pmod{341}$, but $341 = 31 \cdot 11$ is composite. So just testing whether (1) holds with $a = 2$ isn't enough to weed out all composite numbers. Well, ok, so maybe it's not enough to just use $a = 2$; let's also test if (1) holds with $a = 3$. In this case we find $3^{340} \equiv 56 \pmod{341}$, and this *does* prove that $n = 341$ is composite. So maybe if we test (1) for both $a = 2$ and $a = 3$, no composite numbers slip by. It turns out this is false (e.g., $n = 1105$ slips past this new combined test). Ok, well, what if we try a random approach: let's just pick $1 < a < n$ at random and test whether (1) holds. We might hope that even if the first few a's are bad, most

values of a will make (1) fail, and so we would have a good random algorithm for proving that a given composite number really is composite.

It turns out that this doesn't work. According to a result of Alford, Granville, and Pomerance, there are infinitely many composite n for which (1) holds for every a coprime to n. (Such n are called *Carmichael numbers*.) Nevertheless, this approach *almost works*. By inserting just a few additional ideas into the mix, one can prove the following theorem:

Theorem 1. *There is a good (in our technical sense) random algorithm which, given a composite integer n, either reports 'composite' or 'failed to prove composite,' and reports 'composite' at least 75% of the time (for any given composite input n).*

Here is one such algorithm, which (in various guises) can be attributed to any of Miller, Rabin, and Selfridge. Suppose that n is an odd prime and that a is an integer coprime to n. Write $n-1 = 2^k q$, where $k \geq 1$ and q is odd. It is easy to check that one has the factorization

$$a^{n-1} - 1 = (a^q - 1)(a^q + 1)(a^{2q} + 1) \cdots (a^{2^{k-1}q} + 1).$$

(If you doubt this, just start multiplying out the right-hand product.) Since n is prime, Fermat's little theorem tells us that n divides $a^{n-1} - 1$, and so n must divide one of the factors on the right-hand side. So either

$$a^q \equiv 1 \pmod{n} \quad \text{or, for some } 0 \leq j < k, \quad a^{2^j q} \equiv -1 \pmod{n}. \tag{2}$$

Suppose now that n is composite. It can be shown (though we don't do it here) that (2) fails for at least 3/4 of all choices of $1 \leq a < n$, in accord with the statement of Theorem 1. Notice that testing whether (2) holds is easy; we can compute $a^q \pmod{n}$ by our repeated squaring method and then continue squaring the result modulo n to get $a^{2q}, a^{2^2 q}$, etc.

So suppose you have an n that you suspect is prime. You can run it through this algorithm with 50 random choices of a. If each time the algorithm fails to prove n composite, then either n is prime or you've been very unlucky: you've witnessed an event of probability at most 1 in $4^{50} = 2^{100}$. This is probably enough to convince you that n is prime, but it isn't a proof.

Proving primality

Suppose we are morally certain n is prime, because repeated application of one of the random algorithms alluded to above failed to prove n composite.

If we know the factorization of $n-1$, it is not too hard to take the last step and prove n prime:

Theorem 2 (Lucas)**.** *Suppose $n-1$ is completely factored. Moreover, suppose that for the integer a, we have*

$$a^{n-1} \equiv 1 \pmod{n},$$

but for every prime q dividing $n-1$, we have

$$a^{(n-1)/q} \not\equiv 1 \pmod{n}.$$

Then n is prime.

Remark. Before we prove this, we should remark that if n is actually prime, then the conditions on a in this theorem amount to requiring that a be a generator for the multiplicative group $\mathbb{F}_n^\times$. We've seen in class that there are $\phi(n-1)$ generators of $\mathbb{F}_n^\times$, and in analytic number theory one shows that $\phi(n-1)/(n-1)$ is not too small. Thus, choosing a at random from $[1, n-1]$, we're likely to stumble across an a that works before too long. (More precisely: it can be shown that we don't expect to have to make more than a constant multiple of $\log\log n$ choices of a before we find one that works. Note that $\log\log n$ grows very slowly.)

Proof. Let l represent the order of $a \pmod n$, i.e., the order of a in the group $(\mathbb{Z}/(n))^\times$. From the first congruence in the theorem, we see that l divides $n-1$. From the second (non)congruence of the theorem, we know that for each prime q dividing $n-1$, it's not the case that l divides $(n-1)/q$. This is only possible (by unique factorization) if $l = n-1$. But the order of any element is at most the size of the ambient group. So there must be at least $n-1$ units in $(\mathbb{Z}/(n))^\times$. This means that $1, 2, \ldots, n-1$ must all be units in $\mathbb{Z}/(n)$. But then n is coprime to all integers $1 \leq d < n$, which means that n is prime. □

What if we only know part of the factorization of $n-1$?

Theorem 3 (Pocklington)**.** *Suppose $n-1 = FR$, where F is completely factored. Moreover, suppose that for the integer a, we have*

$$a^{n-1} \equiv 1 \pmod n, \tag{3}$$

but for all primes q dividing F,

$$\gcd(a^{(n-1)/q} - 1, n) = 1. \tag{4}$$

Then every prime p dividing n satisfies $p \equiv 1 \pmod F$.

Proof. Let p be a prime dividing n. Since p divides n, we can read (3) modulo p to find that

$$(a^R)^F = a^{RF} = a^{n-1} \equiv 1 \pmod p.$$

The equality (4) guarantees that for each prime q dividing F, we have

$$(a^R)^{F/q} = a^{RF/q} = a^{(n-1)/q} \not\equiv 1 \pmod p.$$

Reasoning as in the proof of Theorem 2, we see that a^R has order F modulo p. But the order of a^R must divide $p-1$. So $p \equiv 1 \pmod F$. □

Suppose now that $F \geq \sqrt{n}$. (So we can factor a portion of n of length at least about half the length of n.) Then Theorem 3, for the right choice of a, will prove that all of the prime factors p of n satisfy $p \equiv 1 \pmod F$. But then $p \geq 1 + F > \sqrt{n}$. So n only has prime factors exceeding its square root, so it must be prime!

We close with an application of Lucas's Theorem 2. For $k \geq 0$, define $F_k = 2^{2^k} + 1$. (These are called the *Fermat numbers*.) Thus $F_0 = 3, F_1 = 5, F_2 = 17, F_3 = 257, F_4 = 65537$. All these numbers happen to be prime, and Fermat conjectured that F_k is prime for every k. However, Euler showed that

$$F_5 = 641 \cdot 6700417.$$

It is now known that F_k is composite for all $5 \leq k \leq 32$, and it is widely conjectured that F_k is in fact composite for every $k \geq 5$.

How do we test if a large Fermat number is prime? Notice that these numbers get very large very quickly, each being essentially the square of the previous. So we need an efficient algorithm if we hope to get anywhere at all.

Theorem 4 (Pépin)**.** *For $k \geq 1$, the number F_k is prime if and only if*

$$3^{(F_k-1)/2} \equiv -1 \pmod{F_k}.$$

Proof. Let $k \geq 1$ and set $n = F_k$. Notice that $q = 2$ is the only prime dividing $n - 1$. We have that $3^{(n-1)/2} \equiv -11 \pmod{n}$, so that $3^{n-1} \equiv (-1)^2 \equiv 1 \pmod{n}$. This verifies the condition's of Theorem 2 for the integer $a = 3$, so that n is prime.

Suppose conversely that n is prime. It's easy to prove (e.g., by induction on k) that for $k \geq 1$, we have $n \equiv 5 \pmod{12}$. But 3 is not a square modulo any prime from the residue class 5 (mod 12). (This is a special case of the law of quadratic reciprocity from elementary number theory.) Hence $3^{(n-1)/2} \equiv -1 \pmod{n}$, which is exactly the congruence of Pépin's theorem. □