Lecture 7: Congruences, Part III

William Stein

Math 124 HARVARD UNIVERSITY **Fall 2001**

Key Ideas

1. How to solve $ax \equiv 1 \pmod{n}$ efficiently.
2. How to compute $a^m \pmod{n}$ efficiently.
3. A probabilistic primality test.

1 How to Solve $ax \equiv 1 \pmod{n}$

Let $a, n \in \mathbb{N}$ with $\gcd(a, n) = 1$. Then we know that $ax \equiv 1 \pmod{n}$ has a solution. How can we find x?

1.1 More About GCDs

Proposition 1.1. *Suppose $a, b \in \mathbb{Z}$ and $\gcd(a, b) = d$. Then there exists $x, y \in \mathbb{Z}$ such that*

$$ax + by = d.$$

I won't give a formal proof of this proposition, though there are many in the literature. Instead I will show you how to find x and y in practice, because that's what you will need to do in order to solve equations like $ax \equiv 1 \pmod{n}$.

Example 1.2. Let $a = 5$ and $b = 7$. The steps of the Euclidean gcd algorithm are:

$$\begin{aligned}
\underline{7} &= 1 \cdot \underline{5} + \underline{2} & \text{so } \underline{2} &= \underline{7} - \underline{5} \\
\underline{5} &= 2 \cdot \underline{2} + \underline{1} & \text{so } \underline{1} &= \underline{5} - 2 \cdot \underline{2} = 3 \cdot \underline{5} - 2 \cdot \underline{7}
\end{aligned}$$

On the right, we have written each partial remainder as a linear combination of a and b. In the last step, we write $\gcd(a, b)$ as a linear combination of a and b, as desired.

That example wasn't too complicated, next we try a much longer example.

Example 1.3. Let $a = 130$ and $b = 61$. We have

$$\begin{aligned}
\underline{130} &= 2 \cdot \underline{61} + \underline{8} & \text{so } \underline{8} &= \underline{130} - 2 \cdot \underline{61} \\
\underline{61} &= 7 \cdot \underline{8} + \underline{5} & \text{so } \underline{5} &= -7 \cdot \underline{130} + 15 \cdot \underline{61} \\
\underline{8} &= 1 \cdot \underline{5} + \underline{3} & \text{so } \underline{3} &= 8 \cdot \underline{130} - 17 \cdot \underline{61} \\
\underline{5} &= 1 \cdot \underline{3} + \underline{2} & \text{so } \underline{2} &= -15 \cdot \underline{130} + 32 \cdot \underline{61} \\
\underline{3} &= 1 \cdot \underline{2} + \underline{1} & \text{so } \underline{1} &= 23 \cdot \underline{130} - 49 \cdot \underline{61}
\end{aligned}$$

Thus $x = 130$ and $y = -49$.

Remark 1.4. For our present purposes it will always be sufficient to find one solution to $ax + by = d$. In fact, there are always infinitely many solutions. If x, y is a solution to

$$ax + by = d,$$

then for any $\alpha \in \mathbb{Z}$,

$$a\left(x + \alpha \cdot \frac{b}{d}\right) + b\left(y - \alpha \cdot \frac{a}{d}\right) = d,$$

is also a solution, and all solutions are of the above form for some α.

It is also possible to compute x and y using PARI.

```
? ?bezout
bezout(x,y): gives a 3-dimensional row vector [u,v,d] such that
             d=gcd(x,y) and u*x+v*y=d.
? bezout(130,61)
%1 = [23, -49, 1]
```

1.2 To solve $ax \equiv 1 \pmod{n}$

Suppose $\gcd(a, n) = 1$. To solve

$$ax \equiv 1 \pmod{n},$$

find x and y such that $ax + ny = 1$. Then

$$ax \equiv ax + ny \equiv 1 \pmod{n}.$$

Example 1.5. Solve $17x \equiv 1 \pmod{61}$. First, we use the Euclidean algorithm to find x, y such that $17x + 61y = 1$:

$$\begin{aligned}
\underline{61} &= 3 \cdot \underline{17} + \underline{10} & \text{so } \underline{10} &= \underline{61} - 3 \cdot \underline{17} \\
\underline{17} &= 1 \cdot \underline{10} + \underline{7} & \text{so } \underline{7} &= -\underline{61} + 4 \cdot \underline{17} \\
\underline{10} &= 1 \cdot \underline{7} + \underline{3} & \text{so } \underline{3} &= 2 \cdot \underline{61} - 7 \cdot \underline{17} \\
\underline{3} &= 2 \cdot \underline{3} + \underline{1} & \text{so } \underline{1} &= -5 \cdot \underline{61} + 18 \cdot \underline{17}
\end{aligned}$$

Thus $x = 18$ is a solution to $17x \equiv 1 \pmod{61}$.

2 How to Compute $a^m \pmod{n}$ Efficiently

As we will see on Friday, a quick method to compute $a^m \pmod{n}$ is absolutely *essential* to public-key cryptography.

Naive Algorithm: Compute $a \cdot a \cdot \cdots \cdot a \pmod{n}$ by repeatedly multiplying by a and reducing modulo m. This is *BAD* because it takes $m-1$ multiplications.

Clever Algorithm: The following observation is the key idea which makes the clever algorithm work. Write $m = \sum_{i=1}^{r} \varepsilon_i 2^i$ with each $\varepsilon_i \in \{0,1\}$, i.e., write m in base 2 (binary). Then

$$a^m = \prod_{\varepsilon_i = 1} a^{2^i} \pmod{n}.$$

It is straightforward to write a number m in binary, as follows: If m is odd, then $\varepsilon_0 = 1$, otherwise $\varepsilon_0 = 0$. Replace m by floor$(\frac{m}{2})$. If the new m is odd then $\varepsilon_1 = 1$, otherwise $\varepsilon_1 = 0$. Keep repeating until $m = 0$.

Example 2.1.
Problem: *Compute the last* 2 *digits of* 6^{91}.

Solution: We compute $6^{91} \pmod{100}$.

i	m	ε_i	$6^{2^i} \bmod 100$
0	91	1	6
1	45	1	36
2	22	0	96
3	11	1	16
4	5	1	56
5	2	0	36
6	1	1	96

As a check, note that $91 = 1011011_2 = 2^6 + 2^4 + 2^3 + 2 + 2^0$. Finally, we have

$$6^{91} = 6^{2^6} \cdot 6^{2^4} \cdot 6^{2^3} \cdot 6^2 \cdot 6 \equiv 96 \cdot 56 \cdot 16 \cdot 36 \cdot 6 \equiv 56 \pmod{100}.$$

Summary of above table: The first column, labeled i, is just to keep track of i. The second column, labeled m, is got by dividing the entry above it by 2 and taking the integer part of the result. The third column, labeled ε_i, simply records whether or not the second column is odd. The forth column is computed by squaring, modulo 100, the entry above it.

Some examples in PARI to convince you that powering isn't too difficult:

```
? Mod(17,389)^5000
%13 = Mod(330, 389)
? Mod(2903,49084098)^498494
%14 = Mod(13189243, 49084098)
```

These both take no noticeable time.

3 A Probabilistic Primality Test

Recall,

Theorem 3.1. *A natural number p is prime if and only if for* every $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Thus if $p \in \mathbb{N}$ and, e.g., $2^{p-1} \not\equiv 1 \pmod{p}$, then we have proved that p is *not* prime. If, however, $a^{p-1} \equiv 1 \pmod{p}$ for a couple of a, then it is "highly likely" that p is prime. I will not analyze this probability here, but we might later in this course.

Example 3.2. Let $p = 323$. Is p prime? Let's compute 2^{322} modulo 323. Making a table as above, we have

i	m	ε_i	$2^{2^i} \bmod 323$
0	322	0	2
1	161	1	4
2	80	0	16
3	40	0	256
4	20	0	290
5	10	0	120
6	5	1	188
7	2	0	137
8	1	1	35

Thus

$$2^{322} \equiv 4 \cdot 188 \cdot 35 \equiv 157 \pmod{323},$$

so 323 is not prime. In fact, $323 = 17 \cdot 19$.

It's possible to prove that a large number is composite, but yet be unable to (easily) find a factorization! For example if

$$n = 95468093486093450983409583409850934850938459083,$$

then $2^{n-1} \not\equiv 1 \pmod{n}$, so n is composite. This is something one could verify in a reasonable amount of time by hand. (Though finding a factorization by hand would be very difficult!)

3.1 Finding large numbers that are probably prime

```
? probprime(n, a=2) = Mod(a,n)^(n-1) == Mod(1,n)
? x = 0948609348698406983409580934859034509834095809348509834905809345
%36 = 948609348698406983409580934859034509834095809348509834905809345
? for(i=0,100,if(probprime(x+2*i,2),print(i)))
27
? p = x + 2*27
%37 = 948609348698406983409580934859034509834095809348509834905809399
? probprime(p,3)
%39 = 1
```