11.1 Overview

1. Multiplication, Division and Exponentiation mod \( m \)
2. Fermat’s Little Theorem and Primality Testing

11.2 Counting Steps

11.2.1 Multiplication

Let us define \( M(n) \) as the number of steps it takes to multiply 2 \( n \)-bit numbers. This will be useful as a unit operation, when we consider other arithmetic operations.

In grade school, we learn an algorithm which gives \( M(n) = O(n^2) \).

In grad school, we learn an algorithm which gives \( M(n) = O(n^{1+\epsilon}) \) for arbitrary \( \epsilon \). Stephen Cook presented such an algorithm in his PhD Thesis (cf. [http://cr.yp.to/bib/1966/cook.html](http://cr.yp.to/bib/1966/cook.html)). In the same work, Cook also showed that \( O(n) \) time cannot be achieved on a certain restricted computational model. Later, Schoenhage and Strassen found an \( O(n \log n \log \log n) \) algorithm.

It is an open question as to whether there exists an algorithm (in an unrestricted model) which gives \( M(n) = O(n) \).

11.2.2 Division

Note that division of two \( n \)-bit numbers also takes \( M(n) \) steps - any multiplication algorithm has a corresponding division algorithm.

11.2.3 Exponentiation

Exponentiation, perhaps unsurprisingly, takes longer.

11.2.3.1 First Illuminating example

Imagine you are asked to compute the \( m \)th power of 2, where \( m \) is an \( n \)-bit number. Of course, the answer is very simple:

\[
\begin{array}{c}
1 \\
0 \ldots 0 \\
m \text{ zeroes}
\end{array}
\]

But that could be \( 2^n \) 0s! It would take an exponential time in the size of the input (which is \( n = \log m \)) just to write out the answer.
A nice way around this unfortunate observation to do exponentiation mod an $n$-bit number - now the input and output are the same size.

**Claim:** Given $n$-bit numbers $a, b, m$, we can compute $a^b \pmod{m}$ efficiently, i.e. in $p(n)$ time for some polynomial $p$. In fact, we will show exponentiation is $O(nM(n))$.

### 11.2.3.2 Second Illuminating Example

Let $b = 13$. In binary, $b = 2^{1101}$.

To compute $a^{1101} \pmod{m}$, we start with $a$ raised to the power 1 (our input) and calculate $a$ to higher exponents by perform repeated multiplications, reducing mod $m$ after each operation.

\[
a^1 \rightarrow a^2 \rightarrow a^3 \rightarrow a^6 \rightarrow a^{12} \rightarrow a^{13}
\]

In binary, the exponents are

\[
1 \rightarrow 10 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 1101
\]

### 11.2.3.3 General Algorithm

Set $X = a^1$. For $i$ from 0 to $n - 1$:

1. Square $X$. $[X = X^2 \pmod{m}]$
2. If the $i$th bit of the binary expansion of $b$ is a 1
   
   (a) Multiply by $a$. $[X = aX \pmod{m}]$

Since we are working mod $m$, $X$ is always an $n$-bit number. Thus, each multiplication takes $M(n)$ steps. The loop is iterated $n$ times, and there are at most two multiplications per iteration, so the runtime is $O(nM(n))$, as claimed above.

### 11.3 Fermat

Fermat thought about this. He came up with some cool beans. Now we will rediscover them.

Fermat noticed:

\[
\begin{array}{cccccccccccc}
2^3 \pmod{n} & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\
\hline
2^2 \pmod{n} & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2
\end{array}
\]

The table seems to suggest, given natural numbers $a, n$,

\[
a^n \pmod{n} = a \quad \Leftrightarrow \quad n \text{ is prime}
\]

This would be a wonderful theorem to have, since it gives a polynomial time test for primality: given $n$, return PRIME iff $2^n = 2 \pmod{n}$. Unfortunately, only one direction is true.
11.3.1 \iff
Fermat showed the left implication with a clever application of the Binomial Theorem. Given prime \( p \),
\[
2^p \equiv_p (1+1)^p \equiv_p \sum_{k=0}^{p} \binom{p}{k}
\]
Notice that for positive \( i < p \), \( \binom{p}{i} = \frac{p!}{i!(p-i)!} \) has no factor of \( p \) in the denominator, but it has one in the numerator. We know that \( \binom{p}{i} \) is an integer, so its prime factorization will contain a \( p \) in it. That is, \( p | \binom{p}{i} \), which gives
\[
\binom{p}{i} \equiv \begin{cases} 
1 \pmod{p} & i = 1 \\
0 \pmod{p} & 0 < i < p \\
1 \pmod{p} & i = p 
\end{cases}
\]
Therefore \( 2^p \equiv 2 \pmod{p} \) for \( p \) prime.

More generally, \((a + 1)^p = \sum_{k=0}^{p} a^k \binom{p}{k}\). By the same factoring argument, \( \binom{p}{k} \) is divisible by \( p \) for \( 0 < k < p \), so \( a^k \binom{p}{k} \equiv 0 \pmod{p} \), and all those terms drop out of the sum. So
\[
(a + 1)^p \equiv a^p + 1 \pmod{p}
\]
To complete the proof, we need \( a^p + 1 \equiv a + 1 \pmod{p} \), or equivalently \( a^p \equiv a \pmod{p} \). We seem to be back where we started, but instead of trying to prove the theorem for \( a + 1 \), we have a smaller number \( a \). So, we can fix the proof by inducting on \( a \).

Equivalently, assume for contradiction that \( a^p \not\equiv a \pmod{p} \) for some natural \( a \). Then there is some smallest counterexample \( x \). \( x > 1 \), since \( 0^p \equiv 0 \pmod{p} \) and \( 1^p \equiv 1 \pmod{p} \). Thus \( (x-1)^p \equiv x \pmod{p} \). However, above we showed \((x-1)^p \equiv x \pmod{p} \Rightarrow x^p \equiv x \pmod{p} \). Contradiction. Thus \( a^p \equiv a \pmod{p} \) for all natural \( a \).

11.3.2 \n\nThere is a very nice counterexample which happens to break lots of primality testing algorithms: 1729.

11.3.2.1 Historical Aside

Srinivasa Ramanujan was an Indian mathematical savant at the turn of the 20th century. His friend Hardy, also a famous mathematician, had this anecdote to relate:

“I remember once going to see [Ramanujan] when he was lying ill at Putney. I had ridden in taxi cab number 1729 and remarked that the number seemed to me rather a dull one, and that I hoped it was not an unfavorable omen. ‘No,’ he replied, ‘it is a very interesting number; it is the smallest number expressible as the sum of two cubes in two different ways.’”

In fact, \( 1729 = 12^3 + 1^3 = 10^3 + 9^3 \).
11.3.3 Primality Testing

$$1729 = 7 \times 13 \times 19$$

but

$$2^{1729} \equiv 2 \pmod{1729}$$

Thus, 1729 breaks our naive primality test. However, perhaps we were unlucky in our choice of 2. Is it the case that for composite \( n \) there exists some \( a \) such that \( a^n \neq a \pmod{n} \)?

Unfortunately, no. It turns out that for all integer \( a \),

$$a^{1729} \equiv a \pmod{1729}$$

Numbers \( n \) for which \( a^n \equiv a \pmod{n} \) for any \( a \) are called pseudoprimes, or Carmichael numbers. 1729 happens to be the third smallest Carmichael number.

11.3.4 Beyond Fermat

We can consider variations on the exponentiation theme.

11.3.4.1 Notice that \( a^{p-1} \equiv 1 \pmod{p} \) for \( p \) prime and \( a \) coprime with \( p \). In fact, this is an equivalent form of Fermat’s Little Theorem.

Given \( a^p \equiv a \pmod{p} \), if \( a \) is coprime with \( p \) then \( a^{-1} \) exists and we can multiply on both sides by \( a^{-1} \) to get \( a^p \equiv a \pmod{p} \).

Given \( a^{p-1} \equiv 1 \pmod{p} \) for \( a \) is coprime with \( p \), then we can multiply on both sides by \( a \) to get \( a^p \equiv a \pmod{p} \) for \( a \) coprime with \( p \). For \( a \) not coprime with \( p \), \( a \) must be a multiple of \( p \), so that \( a^p \equiv 0 \equiv a \pmod{p} \).

So, again, all primes will pass this test, but not all numbers which pass this test are primes. Particularly, \( n = 1729 \) gives \( 2^{1728} \equiv 1 \pmod{1729} \)

11.3.4.2 If \( n \) is odd, then \( (n-1)/2 \) is an integer. Let’s look at \( 2^{(n-1)/2} \pmod{n} \). If \( n \) is prime, then we know that \( (2^{(n-1)/2})^2 \equiv 1 \pmod{n} \), so \( 2^{(n-1)/2} \) is either 1 or \(-1\) modulo \( n \).

But, once again,

$$2^{1728/2} = 1 \pmod{1729}$$

However, notice that 1728 is not just divisible by 2, it is in fact divisible by \( 2^6 \). Can we do something with \( 2^{1728/64} \pmod{1729} \)? Our very own Gary Miller pursued this line of thought and eventually came up with the famous Miller-Rabin probabilistic primality test, which we will talk about next time.