Advanced Algorithms	Lecture 1 : March 6, 1995
Lecturer: Johan Håstad	*Scribe: Christer Berg*

1.1 Introduction

In this course we will study algorithms from various fields of mathematics and theoretical computer science. The goal of course being as efficient methods as possible, our main interest will be running time as measured in the number of operations as a function of the input size. The computational model will be an ordinary computer with limited wordsize.

1.2 Primality testing

Given a (large, for example, more than 100 decimal digits) number is it a prime? Let N be the given number and assume that the length of N is n bits, that is, $N \approx 2^n$.

The naive approach to determine if N is prime is to try and divide N by all numbers smaller than $\sqrt{N}$. Since there are $\sqrt{N} = \sqrt{2^n}$ numbers to try, this will take time $O(2^{n/2})$, which is infeasible if n is large.

A better idea is to make use of Fermat's little theorem which says that if p is prime then $a^{p-1} \equiv 1 \bmod p$ for all a such that $1 \leq a \leq p-1$. However, before we can use this theorem, there are two problems that we have to consider:

1. Is this method efficient?
2. Is the converse of Fermat's theorem true, that is, does $a^{N-1} \equiv 1 \bmod N$ imply that N is prime?

To begin with the first item, the question is how many operations we need in order to compute $a^{N-1} \bmod N$ where $N \approx 2^n$. There are two tricks we can use to do this. First, note that we need only remember partial results modulo N. Second, instead of multiplying a together $N-1$ times, we can square previous results. In general, we will produce powers by either squaring previous result mod N or squaring and then multiplying by a mod N. If we write the exponent in binary this becomes clear, since if we scan the bits in the exponent from most to least significant bit, a zero means that the exponent becomes twice as big, that is, our previous result is squared. If we read a one, the exponent is doubled and a one is added which is to say that we shall square our previous result and then multiply by a. See the following

Example 1 *Suppose we want to compute* $3^{13} \bmod 17$. *Writing* 13 *in binary we get* 1101*, that is,* $13 = 2^3 + 2^2 + 2^0$. *Hence, we want to compute* $3^{13} = 3^{2^3+2^2+2^0} \bmod 17$ *and this can be done as follows.*

$$\begin{aligned} 3 \cdot 1^2 &\equiv 3 \bmod 17 \\ 3 \cdot 3^2 &\equiv 10 \bmod 17 \\ 10^2 &\equiv 15 \bmod 17 \\ 3 \cdot 15^2 &\equiv 12 \bmod 17 \end{aligned}$$

Since N has n binary digits, $a^{N-1} \bmod N$ can be computed with $2n$ multiplications of n-digit numbers together with n modular reductions. Since one multiplication and one modular reduction of n-bit numbers each takes $O(n^2)$ bit operations, the total time for modular exponentiation is $O(n^3)$. Hence, the answer to the first question is yes, since we consider this complexity as quite efficient.

Next for the second question above, that is, is the converse of Fermat's theorem true? Well, if we do have that $a^{N-1} \equiv 1 \bmod N$ for all a between 1 and $p-1$ then N must be prime since in this case $\gcd(a, N) = 1$ for $1 \leq a \leq N-1$ which is just to say that N is prime. But now assume that we have $a^{N-1} \equiv 1 \bmod N$ for all *a relatively prime* to N, is it then the case that N must be prime? No it need not; there are numbers for which $a^{N-1} \equiv 1 \bmod N$ for all a such that $\gcd(a, N) = 1$. These numbers are known as *Carmichael numbers*, the smallest of which is 561. It follows that we can not use Fermat's little theorem as it is to separate primes from composites, but we shall see that it is possible to change it slightly such that it, at least for most a's, reveals composites – even Carmichael numbers.

Consider the Carmichael number 561, which is equal to $3 \cdot 11 \cdot 17$. Instead of looking at a number $a \bmod 561$, we will look at $a \bmod 3$, $a \bmod 11$ and $a \bmod 17$. Let $N = p_1 p_2 \cdots p_r$. The *Chinese remainder theorem* states that the equation

$$\begin{aligned} x &\equiv a_1 \bmod p_1, \\ x &\equiv a_2 \bmod p_2, \\ &\vdots \\ x &\equiv a_r \bmod p_r, \end{aligned}$$

has a unique solution $\bmod N$.

This means that if we consider $(a_1, a_2, a_3) = (a \bmod 3, a \bmod 11, a \bmod 17)$ the Chinese remainder theorem assures that this will correspond to a unique number $a \bmod 561$.

Example 2 *Let $N = 15 = 3 \cdot 5$. Then each number $a \bmod 15$ can be represented as $(a \bmod 3, a \bmod 5)$, as follows.*

$$\begin{array}{lll} 0 = (0,0), & 5 = (2,0), & 10 = (1,0), \\ 1 = (1,1), & 6 = (0,1), & 11 = (2,1), \\ 2 = (2,2), & 7 = (1,2), & 12 = (0,2), \\ 3 = (0,3), & 8 = (2,3), & 13 = (1,3), \\ 4 = (1,4), & 9 = (0,4), & 14 = (2,4). \end{array}$$

When computing, for example, $8 \cdot 14$ we want this to equal $112 \equiv 7 \bmod 15$ – is this still the case when using the above representation? Well, we get

$$8 \cdot 14 = (2,3) \cdot (2,4) = (2 \cdot 2, 3 \cdot 4) = (4, 12) \equiv (1,2) = 7.$$

1.3 The Miller-Rabin primality test

Let $N - 1 = 2^s t$ where t is odd. For a given a we compute

$$\begin{aligned} u_0 &\equiv a^t \bmod N \\ u_{i+1} &\equiv u_i^2 \bmod N \\ u_s &\equiv a^{2^s t} = a^{N-1} \bmod N \end{aligned}$$

Before, when using Fermat's little theorem, we said that N probably was prime if $u_s \equiv 1 \bmod N$. Now we shall declare N as probably prime if either $u_0 \equiv 1 \bmod N$ or if $u_r \equiv -1 \bmod N$ for some $r < s$.

Assume that N is prime. Then, by the above procedure, we will always declare N as probably prime. This follows since we know that $u_s \equiv 1 \bmod N$, so either we have that $u_0 \equiv 1 \bmod N$ or else there is a last u_i such that $u_i \not\equiv 1 \bmod N$ and $u_{i+1} \equiv u_i^2 \equiv 1 \bmod N$. But if N is prime, the equation $x^2 \equiv 1 \bmod N$ has only the solutions $x = \pm 1$ and so we must have $u_i \equiv -1 \bmod N$ which means that N will be declared as probably prime.

On the other hand, if N is not prime, we want to prove that half of the a's will reveal N as not prime. The intuition why this is the case can be described as follows. Suppose that $N = p_1 p_2 \cdots p_r$ and that we pick an $a \bmod N$ at random. But, using the Chinese remainder theorem, we see that picking a random $a \bmod N$ is just independently picking a random $a_1 \bmod p_1$, a random $a_2 \bmod p_2$, and so on. Now, in order for $a \bmod N$ to equal -1 we must, by the Chinese remainder theorem, have that

$$\begin{aligned} a_1 &\equiv -1 \bmod p_1 \\ a_2 &\equiv -1 \bmod p_2 \\ &\vdots \\ a_r &\equiv -1 \bmod p_r, \end{aligned}$$

and that this occurs when all the a_i, $1 \le i \le r$, are picked independently at random, seems very unlikely.

Example 3 *Let's see how the Miller-Rabin test works in the case that N is the Carmichael number $561 = 3 \cdot 11 \cdot 17$ and a is equal to 2. In this case $561 - 1 = 16 \cdot 35$ so that $s = 4$ and $t = 35$.*

	mod3	mod11	mod17	mod561
u_0	-1	-1	8	263
u_1	1	1	13	166
u_2	1	1	-1	67
u_3	1	1	1	1
u_4	1	1	1	1

Here, we have that no row consists of only -1 modulo the three primes in 561. Hence, we never get $-1 \bmod 561$ and since also $u_0 \not\equiv 1 \bmod 561$ the Miller-Rabin test will correctly say that 561 is composite.