01204211 Discrete Mathematics
Lecture 16: Primality testing (1)

Jittat Fakcharoenphol

October 12, 2015

Integers

This is the second major area that we shall study in this course: integers and their properties. This area is called **number theory.** We will see many properties of integers and their applications that include data encoding techniques and data encryption.
Most techniques depend on having large primes. So, we start this area by focus on the primality testing algorithm. This is also the first algorithm that we considered in this class as well.

Brute-force division

FUNCTION CheckPrime2(n)
1. $k \leftarrow 2$
2. WHILE $k \leq \sqrt{n}$ DO
3. IF k divides n THEN
4. RETURN **false**
5. ENDIF
6. $k \leftarrow k + 1$
7. ENDWHILE
8. RETURN **true**

The WHILE loop in CheckPrime2 runs for at most $\sqrt{n}$ times. This improves over the original algorithm that uses roughly n rounds. While this is a big improvement, it is usually not good enough when we consider a typical usage of the algorithm that we need, where we need to check if a 1000-digit number is a prime.

Running time analysis: the O-notation

- We shall study the running time of various algorithms in this section. Since we will not be very precise, keeping tracks of all details, we will use the O-notation when we talk about the running time.
- We will informally use the notation. When we say that function $f(n)$ is $O(g(n))$, we means that the growth of $f(n)$ is at most that of $g(n)$. This means that the largest terms in $f(n)$ is not larger than that of $g(n)$.
- **Examples:**
  - $n^2 + 100n = O(n^2)$
  - $\sqrt{n} + n = O(n)$
  - $2^n + 10000 + 100000n^{10} = O(2^n)$
  - $5 \cdot n^3 + n^2 = O(n^3)$
  - $10n^2 \times 7n + 12n^3 \times 75n^2 = O(n^3) + O(n^5) = O(n^5)$
- Note that it is also true that $n^2 = O(n^3)$.

Polynomial running times

Let's discuss CheckPrime2's running time.
We usually think about the running time as a function on the size of the input. When we want to sort n numbers, the size of the input is n. However, when dealing with big numbers (i.e., those much larger than directly manipulatable in the CPU), you cannot manipulate them in constant time. In this case, we usually count the number of bits as the size of the input.
Since n is the value of the input, to keep this integer in computer memory, you need at least $\log_2 n$ bits[1]. This will be the size of the input that we shall consider.

Let $m = O \log n$ be the number of bits of n. The algorithm that runs in time $O(\sqrt{n})$, actually runs in time $O(\sqrt{2^m}) = O(2^{m/2})$, an exponential running time. This means that the algorithm does not scale very well, as the size of the input increases.
In this case, we want a more efficient algorithm, i.e., it runs in time in the polynomial of m.

[1] We shall use logarithm base 2 in this part of the course.

Running time: integer operations

When we have two m-bit integers a and b The running times for

- Addition and subtraction: $O(m)$, and
- Multiplication and division: $O(m^2)$.

Examples:

$$
\begin{array}{r}
1000\ 1110\ 1011\ 0110 \\
+ \\
0010\ 0111\ 0110\ 1111 \\
\hline
1011\ 0110\ 0010\ 0111
\end{array}
$$

$$
\begin{array}{r}
1000\ 1110 \\
\times \\
0010\ 0111 \\
\hline
1000\ 1110 \\
1\ 0001\ 110 \\
10\ 0011\ 10 \\
1\ 0001\ 110 \\
= \\
1\ 0101\ 1010\ 0010
\end{array}
$$

This lecture covers

- basic definitions related to division and modulo operation
- prime factorization
- a fundamental theorem stating a fact related to prime numbers called the Fermat's Little Theorem

Definitions: divisibilty

Let's start with basic definitions.

- We say that "a divides b", "b is divisible by a", or "b is a multiple of a" if there exists an integer k such that $b = ka$. In this case, we write
$$a|b.$$
- If it's not the case, we write $a \not| b$.
- When a does not divides b, there is a remainder. We say that r is a remainder of dividing b by a if $0 \leq r < a$ and there exists integer k such that $b = ka + r$. We also write
$$r = a \mod b.$$

  - $10 \mod 3 = 1$, $10 \mod 2 = 0$, $10 \mod 15 = 10$
  - $-10 \mod 3 = 2$, $-10 \mod 15 = 5$

The modulo operation

For integers a and b and positive integer q, we have

- $(a+b) \mod q = ((a \mod q) + (b \mod q)) \mod q$
- $(a-b) \mod q = ((a \mod q) - (b \mod q)) \mod q$
- $(ab) \mod q = ((a \mod q) \times (b \mod q)) \mod q$

Examples:

- $(14+7) \mod 5 = ((14 \mod 5) + (7 \mod 5)) \mod 5 = (4+2) \mod 5 = 1$
- $(14 \cdot 7 \cdot 13 \cdot 19) \mod 5 = ((14 \mod 5) \cdot (7 \mod 5) \cdot (13 \mod 5) \cdot (19 \mod 5)) \mod 5 = (4 \cdot 2 \cdot 3 \cdot 4) \mod 5 = 96 \mod 5 = 1$

These facts are really helpful when you try to compute $x \mod y$ when x is very large compared to y and x is a result of many operations of small numbers. In this case, we can keep moduloing intermediate results to keep them under y.

You will prove these properties in your homework.

Definitions: primes

An integer p is a **prime** if $p > 1$ and p has only 4 factors: $1, -1, p$, and $-p$. If a number larger than 1 is not a prime, we say that it is a **composite**.

Prime numbers are very fascinating. There are many facts that have proved about them. E.g., we looked at Euclid's proof that there are infinitely many primes. Here's another one by Euclid:

Theorem: For any positive integer n, there are n consecutive composites.
Proof: Let $m = n + 1$. Consider

$$(m! + 2), (m! + 3), \ldots, (m! + m).$$

Note that these $m - 1 = n$ numbers are composite because for any $1 \leq i \leq m$, $i|m!$, $i|i$, and thus, $i|(m! + i)$. ■

Prime factorization

It is known since the Greeks that if you have a composite n, you can factor it as a product of prime numbers. For example, you can write

$$140 = 2 \times 2 \times 5 \times 7.$$

Not only you can do that, but the Greeks also know that you can do that in only one way (except the permutation of the prime factors). Many proofs we shall introduce later on require this fact, so we shall prove it next.

Theorem: A prime factorization of a positive number larger than 1 is unique.

Proof

We shall prove by contradiction. We also use an argument usually referred to as the "minimal criminal" argument, for which we use the fact that we can pick the minimal element of a subset of positive integers.

Assume that the statement is false, i.e., there exists a positive integer with two prime factorizations. Let n be the smallest integer with that property. I.e., n has at least two prime factorizations:

$$n = p_1 \cdot p_2 \cdots p_r,$$

and

$$n = q_1 \cdot q_2 \cdots q_s.$$

Proof (cont.)

To simplify our proof, we shall make a few assumptions. Note that these assumptions do not change the actual assumption of the theorem. When we make these types of assumption, we usually say that we make assumptions "without loss of generality".

1st assumption. We assume that the two prime factorizations do not share any primes, i.e., the sets $\{p_1, p_2, \ldots, p_r\}$ and $\{q_1, q_2, \ldots, q_s\}$ are disjoint. If this is not the case, we can divide n by a common prime factor p_i to obtain a smaller integer n' with two prime factorizations.

2nd assumption. We also assume that p_1 is the smallest prime factor in both prime factorizations, i.e., $p_1 \leq p_i$ for all $1 \leq i \leq r$ and $p_1 < q_j$ for all $1 \leq j \leq s$. Otherwise, we can switch the roles of prime factorizations $p_1 p_2 \cdots p_r$ and $q_1 q_2 \cdots q_s$.

Proof (cont.)

We shall proceed to show that the assumption leads to a contradiction by proving that there exists an integer n' smaller than n with two prime factorizations.
Let's take p_1 and divide every prime q_j with p_1. Let r_j be each remainder, i.e., for $1 \leq j \leq s$, let

$$r_j = q_j \mod p_1.$$

Since p_1 does not appear in the second prime factorization, it does not divide any q_j; thus, $r_j \neq 0$.
Let

$$n' = r_1 \cdot r_2 \cdots r_s.$$

From this definition, we can obtain one prime factorization of n' by combining all prime factorizations of all r_j.

Proof (cont.)

It is left to show that $n' = r_1 \cdot r_2 \cdots r_s$ has another prime factorization. To do so, we shall prove that $p_1 | n'$.

Since $r_j = q_j \mod p_1$, we can write

$$q_j = k_j \cdot p_1 + r_j,$$

for some integer k_j. Thus, we have that

$$n = (k_1 p_1 + r_1)(k_2 p_1 + r_2) \cdots (k_s p_1 + r_s),$$

which can be written as

$$n = K \cdot p_1 + r_1 \cdot r_2 \cdots r_s,$$

for some integer K. Since $p_1 | n$, we have that $p_1 | r_1 \cdot r_2 \cdots r_s$ (or, $p_1 | n'$).

Proof (cont.)

Since $p_1|n'$, there exists a prime factorization of n' that has p_1 as a factor. To see that this is a different prime factorization, recall that all $r_j < p_1$ (because they are remainders of divisions by p_1). This means that every prime in the first prime factorization is less than p_1, but the second prime factorization has p_1 as a factor. Hence, they are different.

This leads to the contradiction, because we now have a smaller integer n' with more than one prime factorizations. ■

Fermat's Little Theorem

Fermat has another famous theorem which is very useful in number theory. It can be stated as follows.

> Theorem: If p is a prime and a is an integer not divisible by p, we have that
>
> $$a^{p-1} \mod p = 1.$$

To prove this theorem, we shall use the unique prime factorization theorem to prove a short lemma related to divisibility of binomial coefficients first.

Lemma: If p is a prime for any integer $1 \leq k < p$, we have that

$$p \Big| \binom{p}{k}.$$

Proof: Note that

$$\binom{p}{k} = \frac{p(p-1)(p-2)\cdots(p-k+1)}{k!}$$

Since p is in the product in the numerator,

$$p | p(p-1)(p-2)\cdots(p-k+1).$$

Now consider the denominator $k!$. Since it is a product of numbers less than p, its unique prime factorization does not contain p. Hence, the fraction has p as a factor; thus, $p \big| \binom{p}{k}$ as required. ■

Fermat's Little Theorem

We shall prove a different form of the theorem. (Its equivalence to the standard form shall be proved in the next lecture.)

Theorem: If p is a prime and for any integer a, we have that $p|a^p - a$.

Proof

We shall prove by induction on a. Let $P(a)$ be the statement that $p|a^p - a$.

Base case: Since $p|0^p - 0$, $P(0)$ is true.

Proof (cont.)

Inductive step: Assume $P(k)$ is true, we shall prove that $P(k+1)$ is true. From the binomial theorem, we have

$$(k+1)^p - (k+1)$$

$$\begin{aligned} &= \left(k^p + \binom{p}{p-1}k^{p-1} + \binom{p}{p-2}k^{p-2} + \cdots + \binom{p}{1}k + 1\right) \\ &\quad -(k+1) \\ &= (k^p - k) + \left(\binom{p}{p-1}k^{p-1} + \binom{p}{p-2}k^{p-2} + \cdots + \binom{p}{1}k\right). \end{aligned}$$

The first term is divisible by p from the induction hypothesis. The second term is a sum of terms divisible by p from the previous lemma. Thus, $p|(k+1)^p-(k-1)$, implying $P(k+1)$ as required. ■