There is no known polynomial time algorithm to factor integers, not even a probabilistic one. We present in this text an algorithm which, although exponential time, is more efficient than trial division.

1. Birthday paradox

Assume there are 40 students in a classroom. The probability that two among them have the same birthday (assuming none of them was born on February 29th) is

\[
1 - (1 - \frac{1}{365})(1 - \frac{2}{365}) \cdots (1 - \frac{39}{365}) \geq 0.89
\]

This is rather close to 1 although 40 is much less than 365 . . .

We look for a conceptual explanation of this phenomenon.

Assume we have \( p \) balls in a bag, each tagged with a figure from 1 to \( p \). We draw a ball at random and put it back in the bag. We iterate \( n \geq 2 \) times. We now estimate the probability \( P(p, n) \) for the \( n \) drawn balls to be pairwise distinct:

\[
P(p, n) = \prod_{1 \leq i \leq n-1} \left(1 - \frac{i}{p}\right) \leq \prod_{1 \leq i \leq n-1} \exp\left(-\frac{i}{p}\right) = \exp\left(-\frac{n(n-1)}{2p}\right) \leq \exp\left(-\frac{(n-1)^2}{2p}\right)
\]

So if \( n \) is greater than \( 1 + \sqrt{p} \) the probability that the same ball has been drawn more than once is at least \( 1 - \exp(-1/2) > 0.39 \).

If the number of draw is proportional to the square root of the number of balls there is a significant probability to have drawn twice the same ball in the end.

2. Random maps from a finite set to itself

Let \( F \) be a finite set with \( p \) elements. Let \( \mathcal{A}(F) \) be the set of maps from \( F \) to \( F \). Consider the uniform probability measure on \( \mathcal{A}(F) \). Fix an element \( O \) in \( F \).

To every map \( f : F \to F \) we associate the sequence \( x_0 = O, x_{i+1} = f(x_i) \) obtained by iterating \( f \). This is an ultimately periodic sequence: there exist two integers \( \pi_f \geq 1 \) and \( \mu_f \geq 0 \) such that if \( k \geq \mu_f \) then \( x_{k+\pi_f} = x_k \). The smallest such \( \pi_f \) is called the period and the smallest such \( \mu_f \) is called the preperiod.

The sum \( \rho_f = \mu_f + \pi_f \) is a random variable on \( \mathcal{A}(F) \). The probability of the event \( \rho_f \geq n \) is \( P(p, n) \). Could you explain why?

The expectation of \( E(\rho_f) \) satisfies
2 TD3 : AN INTEGER FACTORING ALGORITHM

\[ E(\rho_f) = 1 + \sum_{n \geq 2} P(p, n) \leq \sum_{n \geq 0} \exp(-\frac{n^2}{2p}) \leq 1 + \int_{0}^{\infty} e^{-\frac{x^2}{p}} \, dx \]

\[ = 1 + \sqrt{2p} \int_{0}^{\infty} e^{-x^2} \, dx = 1 + \sqrt{\frac{p\pi}{2}} \]

car \int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}.

3. TWO SIMPLE FACTORING ALGORITHMS

Remind there exists a polynomial time algorithm for primality testing. So factoring integers reduces to the following problem: on input a composite integer \( N \) find a non-trivial divisor \( M \) of \( N \).

Indeed, if the factors \( M \) and \( R = N/M \) are not prime, we rerun the algorithm with \( M \) and \( R \).

An algorithm that finds a non-trivial factor to a given composite integer is called a breaking algorithm. So factoring reduces to iteratively breaking integers.

The simplest factoring algorithm is trial division. To factor \( N \), compute the euclidean division of \( N \) by \( r = 2, 3, 5, 7, 9, 11, 13, 15 \) etc.

If \( N \) is composite you will find a factor \( r \leq \sqrt{N} \).

The complexity of trial division is \( O(N^{1/2+o(1)}) \). This is poorly efficient but useful for small integers.

Pollard has invented an elegant but heuristic method with a better complexity.

We assume to simplify that \( N = pq \) is the product of two distinct primes. We choose a polynomial \( f \) with integer coefficients (one often takes \( f(X) = X^2 + 1 \)) and we consider the sequence with values in \( \mathbb{Z}/N\mathbb{Z} \) defined by \( x = x_0 \) any element in \( \mathbb{Z}/N\mathbb{Z} \) and \( x_{k+1} = f(x_k) \mod N \).

Since \( f(X) \) is a polynomial, the map

\[ f_N : \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{Z}/N\mathbb{Z} \]

\[ x \mod N \longmapsto f(x) \mod N \]

is the set-theoretical product of the two maps

\[ f_p : \mathbb{Z}/p\mathbb{Z} \longrightarrow \mathbb{Z}/p\mathbb{Z} \]

\[ x \mod p \longmapsto f(x) \mod p \]

and
\[
f_q : \mathbb{Z}/q\mathbb{Z} \rightarrow \mathbb{Z}/q\mathbb{Z}
\]

\[
x \mod q \mapsto f(x) \mod q
\]

More precisely we call \( \gamma \) the Chinese isomorphism

\[
\gamma : \mathbb{Z}/N\mathbb{Z} \rightarrow \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}
\]

and we check that \( \gamma \circ f_N = (f_p \times f_q) \circ \gamma \).

Assume now that the maps \( f_p \) and \( f_q \) behave like two independent random maps. In other words assume that \( f_p \) follows a uniform probability law in the set \( A(\mathbb{Z}/p\mathbb{Z}) \) of maps from \( \mathbb{Z}/p\mathbb{Z} \) to itself. Assume also that \( f_q \) follows a uniform probability law and that the random variables \( f_p \) and \( f_q \) are independant.

This is a somewhat crazy assumption since \( f = x^2 + 1 \) is anything but random . . .

Let \( y_k = x_k \mod p \) be the class of \( x_k \) modulo \( p \). Let \( z_k = x_k \mod q \) be the class of \( x_k \) modulo \( q \). One checks that \( y_{k+1} = f_p(y_k) \) and \( z_{k+1} = f_q(z_k) \). The Chinese isomorphism \( \gamma \) sends \( x_k \) onto \( (y_k, z_k) \).

Let \( \pi_p \) and \( \mu_p \) be the period and preperiod of \( f_p \). Let \( \pi_q \) and \( \mu_q \) be the period and preperiod of \( f_q \). We have good reasons to expect that \( \pi_p \) and \( \mu_p \) (that are functions of \( f \) and \( p \)) are \( O(\sqrt{p}) \). We also expect that \( \pi_q \) and \( \mu_q \) are \( O(\sqrt{q}) \). So we have an iterated sequence in \( \mathbb{Z}/N\mathbb{Z} \) whose component modulo \( p \) (resp. \( q \)) has preperiod and period \( O(\sqrt{p}) \) (resp. \( O(\sqrt{q}) \)).

If \( k \) is large enough we thus expect that

\[
gcd(x_k - x_{k+\pi_p}, N) = p
\]

which exhibits a non-trivial factor of \( N \). This is of little help in this form because we do not know \( \pi_p \). However for \( k \) large enough and a multiple of \( \pi_p \) (but not a multiple of \( \pi_q \)) we have

\[
gcd(x_k - 2x_k, N) = p.
\]

Pollard’s algorithm computes iteratively \( x_k = f(x_{k-1}) \) and \( X_k = x_{2k} = f(f(X_{k-1})) \) and the above gcd for \( k = 0, 1, 2, ..., \) until a non-trivial factor of \( N \) shows up.

Heuristically this method finds a non-trivial factor \( p \) in time \( O(p^{1/2+o(1)}) \) that is \( O(\sqrt{N}) \).

4. Questions

(1) Give a cryptographic scheme whose security relies on the difficulty of factoring integers and try to quantify the effect of Pollard’s algorithm on the security of this protocol. What would be a reasonable key length to resist such an attack?

(2) Try to justify the computation of the expectation of \( \rho_f \).

(3) Study the integral \( \int_{-\infty}^{\infty} e^{-x^2} \, dx \), either by computing a numerical approximation or by proving that is is equal to \( \sqrt{\pi} \).

You may want to prove that \( \left( \int_{-\infty}^{\infty} e^{-x^2} \, dx \right)^2 = \pi \). To this end write
\[
\left( \int_{-\infty}^{\infty} e^{-x^2} \, dx \right)^2 = \int_{-\infty}^{\infty} e^{-x^2} \, dx \int_{-\infty}^{\infty} e^{-y^2} \, dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2-y^2} \, dx \, dy
\]

and introduce polar coordinates \( r = \sqrt{x^2 + y^2} \) and the angle \( \theta \).

4) Discuss the relevance of various choices for the polynomial \( f(X) \). You may wonder if \( f(X) = X^2 \) is better or worse than \( f(X) = X^2 + 1 \). Considering that \( (x + 1/x)^2 = x^2 + 1/x^2 + 2 \) what can you say about choosing \( f(X) = X^2 - 2 \)?

5) You may try to express the period \( \pi_N \) and preperiod \( \mu_N \) of \( f_N : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z} \) as functions of \( \pi_p, \mu_p, \pi_q, \mu_q \).

6) Explain why Pollard’s algorithm is very good at finding small prime divisors of large integers and illustrate this method with a simple implementation.