(Integer) Factoring algorithms

There are two families of algorithm to consider:

- (Today) Algorithms for which the complexity depends on the smallest prime factor of \( N \).
- (Next time) Algorithms for which the complexity depends on \( N \) itself.

0 Analytic number theory facts

a The distribution of primes

For \( x \) an integer, let \( \pi(x) \) be the number of prime smaller \( x \).

Theorem 1. (Hadamard, de la Vallée Poussin, 1896).

\[
\pi(x) \sim \int_2^x \frac{dt}{\log(t)} \tag{1}
\]

Observation 2.

- The Riemann Hypothesis is equivalent to the fact that the error term in (1) is \( \mathcal{O}(x^{1/2+\epsilon}) \).
- \( \int_2^x \frac{dt}{\log(t)} = \frac{x}{\log(x)} + \mathcal{O}\left(\frac{x}{\log(x)}\right) \).
- This is hard to prove. In most applications, the so called Chebyshev inequalities are sufficients.

\[
\frac{x}{\log(x)}(1 + o(1)) \leq \pi(x) \leq 2\log(2)\frac{x}{\log(x)}(1 + o(1))
\]

Other properties:

\[
\sum_{p \leq x} \log p = x(1 + o(1)) \quad \sum_{p^\alpha \leq x} \log p = x(1 + o(1))
\]
b Multiplicative structure of random integers

Let $N$ be an integer, and write $N = N_1N_2 \ldots N_k$ with $N_1 \geq \ldots \geq N_k$.

For $a \in \mathbb{N}$, define $P_k(a,X) = \# \{ N : 1 \leq N \leq X, N_k \leq X^{1/a} \}$.

**Theorem 3.** Let $X, a \in \mathbb{N}$, then

$$\frac{P(a,X)}{X} = \rho_k(a) + O \left( \frac{1}{X \log(X)} \right)$$

With:

- $\rho_k(x) = 1$ for $x \in [0,1]$
- $\rho_0 = 0$
- $\rho_k(a) = 1 - \int_1^a (\rho_k(t-1) - \rho_{k-1}(t-1)) dt$

**Corollary 4.** The average value of $\frac{\log(N_1)}{\log(N)}$ is $0.62\ldots$, of $\frac{\log(N_2)}{\log(N)}$ is $0.21\ldots$, and of $\frac{\log(N_3)}{\log(N)}$ is $0.08\ldots$.

c Smooth (Friable) integers

A smooth integer is an integer with only small prime factors.

**Definition 5.**

$$\psi(x,y) = \# \{ n \leq x \mid \text{Largest prime factor of } n \leq y \}.$$ 

For example,

- $\psi(x,2) = \log_2(x)$
- $\psi(x,x^{1/2}) = O(x)$

**Theorem 6.** (A. Hildebrand, 1986) If $x \geq 3$ and $y$ are integers, and there exist $\epsilon > 0$ such that

$$\exp \left( (\log(\log(x)))^{5/3+\epsilon} \right) \leq y \leq x$$

Then, uniformly in $(x,y)$, we have

$$\psi(x,y) = x \rho(u) \left( 1 + O \left( \frac{\log(u)}{\log(y)} \right) \right)$$

Where $u = \frac{\log(x)}{\log(y)}$ and $\rho$ is the Dickman function: $\rho$ continuous and

- $\forall u \in [0,1] \quad \rho(u) = 1$
- $up'(u) + \rho(u-1) = 0$

And we have

**Theorem 7.** For $u \to +\infty$, $\log(u) = -u \log(u)(1 + o(1))$. 
1 "Elementary" methods

a Trial division

If we want to factor $N$, the idea is to try to divide $N$ by every number smaller than $\sqrt{N}$. The complexity is $O(p)$ with $p$ the smallest prime factor of $N$. This method is useful for finding very small prime factors (for example $\leq 10^3$).

b $N^{1/4+\epsilon}$ deterministic

Let $m$ be an integer parameter, which is to be fixed later.

We define $P(X) = X(X+1)\ldots(X+m-1)$ mod $N$. It is computed in $O(m^{1+\epsilon}\log(N))$ along with $P(1), P(m+1), \ldots, P(m(m-1)+1)$ mod $N$ by the interpolation-evaluation algorithm. The gcd of $(P(im+1))_{0 \leq i < m}$ can be computed in time $O(m^{1+\epsilon})$ (see [BCG+17] for a study of both algorithms).

We want $m^2 = N^{1/2}$, i.e., $m = N^{1/4}$. The overall complexity is then $O(N^{1/4+\epsilon})$.

Now this has been computed,

- Either we have, for some $i$, $P(im+1)$, then we can split the interval $[im + 1, (i+1)m]$ into smaller parts.
- Or we have a nontrivial factor somewhere $N = N_1N_2$, we do recursive call with $N_1$ and $N_2$. We make at most $O(\log(N))$ recursive calls.

c Pollard-$\rho$ method
c.1 Iteration of a random mapping

Let $E$ be a finite set, $x_0 \in E$, $f : E \mapsto E$ and $x_{n+1} = f(x_n)$

![Figure 1: Example for $\mathbb{Z}/13\mathbb{Z}$ and $f(x) = x^2 + 1$](image)

**Theorem 8.**

- Has $E$ is finite, the sequence $(x_n)_n$ is ultimately periodic. Namely, $\exists k, t$ such that $\forall n \geq k, x_{n+t} = x_n$. 


• If \((f, x_0)\) are uniform in \(E^E \times E\), then

\[
E(k + t + 1) \sim_{\#E \to \infty} \sqrt{\frac{\pi \cdot \#E}{2}}
\]

**Proposition 9.** \(\exists e \in [k, k + t]\) such that \(x_e = x_{2e}\).

**Proof.** For \(i < j\). \(x_i = x_j \Leftrightarrow i \equiv j \mod t\).

We just need to choose \(e \geq k\) such that \(e \equiv 2e \mod t \Leftrightarrow e \equiv 0 \mod t\).

Actually, it can be proven that \(E(e) = \sqrt{\frac{\pi^5}{288} \cdot \#E}\).

c.2 Application to factoring

**Algorithm 1** Pollard \(\rho\) algorithm

1: \(x \leftarrow 1\)
2: \(y \leftarrow 1\)
3: repeat
4: \(x \leftarrow f(x)\)
5: \(y \leftarrow f(f(y))\)
6: until \(\gcd(x - y, N) \neq 1\)
7: return \(\gcd(x - y, N)\)

Classically, \(f(x) = x^2 + c \mod N\), with \(c \neq 0, 2\).

**what is happening?** We are computing \(x_{2e} = y_e \mod N\). We hope that for some \(p \mid N\), \(e\) is small \((\approx \sqrt{p})\). We will have \(x_e \equiv y_e \mod p\) and then \(p \mid \gcd(x_e - y_e, N)\).

**Example:** \(N = 323, \ f(x) = x^2 + 1\).

<table>
<thead>
<tr>
<th>(X)</th>
<th>(Y)</th>
<th>(\gcd(X - Y, N))</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>226</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>43</td>
<td>316</td>
<td>1</td>
</tr>
<tr>
<td>235</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>316</td>
<td>145</td>
<td>1</td>
</tr>
</tbody>
</table>

Tried with \(N\) = product of 2 random primes \(\in [1, 10^{10}]\), 7 runs of Pollard \(\rho\) algorithm, the number of steps is \(\in [25296, 78153]\).

d Group-Based method

d.1 \(p - 1\) (Pollard)

We fix a bound \(B\), that is to be optimised.

**Example:** \(N = 71080511198562798721, \ B = 3000, \ a = 2, \ X = 4.75 \cdot 10^{19}, \gcd = 7724080517\).
1: Compute $X \leftarrow \prod_{p \leq B} p^\left\lfloor \frac{\log(B)}{\log(p)} \right\rfloor$
2: Sample $a$ uniformly in $\mathbb{Z}/N\mathbb{Z}$
3: if $\gcd(a, N) \neq 1$ then
4:  Return $a$
5: $b \leftarrow \gcd(a^X - 1, N)$
6: if $b \neq 1$ then
7:  Return $b$
8: else
9:  FAIL

What is going on?

Say, $p \mid N$. Then $p \mid \gcd(a^X - 1, N) \iff a^X \equiv 1 \mod p \iff \operatorname{ord}(a \mod p) \mid X$, which is implied by the fact that $p - 1 \mid X$.

Bottom line: we will “Find” $p$ as soon as all prime powers factors of $p - 1$ are $\leq B$.

Cost of the algorithm:

Computing $X$ is linear (in $\log(X)$). Exponentiation $a^X \mod N$ is computed in $O(\log(X)\text{poly}(\log N))$.

$$\log(X) = \sum_{p \leq B} \left\lfloor \frac{\log B}{\log p} \right\rfloor \log p \leq \sum_{p \leq B} \log B = \pi(B) \log(B) = B(1 + o(1))$$

Total cost: $B^{\text{poly} \cdot (\log(N))}$

We expect to find $p \mid N$ for $B \approx p^{0.62\ldots}$.

References