MPRI – Cours 2.12.2

F. Morain

$L[1/2]$ factoring

2019/10/08 and 2019/10/15

The slides are available on http://www.lix.polytechnique.fr/Labo/Francois.Morain/MPRI/2019

I. Basics

Kraitchik (1920): find x s.t. $x^2 \equiv 1 \bmod N, x \neq \pm 1$.

Ex. For $N = 1147$, there are 4 solutions ± 1, ± 371 and $\gcd(371 - 1, N) = 37$.

A general scheme

Step 0: build a prime basis $\mathcal{B} = \{p_1, p_2, \ldots, p_k\}$.

Step 1: find a lot of relations $(R_i)_{i \in I}$: $R_i = \prod_{j=1}^{k} p_j^{a_{i,j}} \equiv 1 \bmod N$

Step 2: find $I' \subset I$ s.t.

$$\prod_{i \in I'} R_i = x^2$$

over $\mathbb{Z}$, which is equivalent to

$$\forall j, \sum_{i \in I'} a_{i,j} \equiv 0 \bmod 2,$$

which is a classical linear algebra problem.

Step 3: x is a squareroot of 1 and with probability $\geq 1/2$, $\gcd(x-1, N)$ is non-trivial.

II. A very naive method

0. Build $\mathcal{B} = \{p_1 = 2, 3, \ldots, p_k\}$.
1. Generate k random relations

$$p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k} \bmod N$$

and hope to factor the residue to get:

$$p_1^{f_1} p_2^{f_2} \cdots p_k^{f_k} \bmod N$$

from which

$$p_1^{e_1 - f_1} p_2^{e_2 - f_2} \cdots p_k^{e_k - f_k} \equiv 1 \bmod N.$$

Store the $(e_i - f_i) \bmod 2$ in the matrix $\mathcal{M}$.

2. Find dependancies relations of $\mathcal{M}$ and deduce solutions of $x^2 \equiv 1 \bmod N$.

Hypothesis:

$$x(e) = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k} \bmod N$$

is a random integer in $[1..N[$.

A numerical example

Let $N = 1147$, $\mathcal{B} = \{2, 3, 5, 7\}$. We compute:

$$(R_1) : 2^{-6} \cdot 3^{10} \cdot 5^6 \cdot 7^8 \equiv 1 \bmod N$$

$$2^1 \cdot 5^5 \cdot 7^2 \equiv 1 \bmod N$$

$$2^1 \cdot 3^6 \cdot 5^5 \cdot 7^5 \equiv 1 \bmod N$$

$$(R_4) : 2^2 \cdot 5^{10} \cdot 7^4 \equiv 1 \bmod N$$

$$2^7 \cdot 3^2 \cdot 5^{-1} \equiv 1 \bmod N$$

$$2^7 \cdot 3^8 \cdot 5^{-1} \cdot 7^3 \equiv 1 \bmod N$$

Combining (R_1) and (R_4), we get:

$$2^{-4} \cdot 3^{10} \cdot 5^{16} \cdot 7^{12} \equiv 1 \bmod N,$$

or $371^2 \equiv 1 \bmod N$ leading to $\gcd(371 - 1, N) = 37$.

De Bruijn's function

Define

$$\psi(x, y) = \#\{z \le x, z \text{ is } y-\text{smooth}\}.$$

Thm. (Candfield, Erdős, Pomerance) $\forall\, \varepsilon > 0$, uniformly in $y \ge (\log x)^{1+\varepsilon}$, as $x \to \infty$

$$\psi(x, y) = \frac{x}{u^{u(1+o(1))}}$$

with $u = \log x / \log y$.

Prop. Let

$$L(x) = \exp\left(\sqrt{\log x \log\log x}\right).$$

For all real $\alpha > 0$, $\beta > 0$, as $x \to \infty$

$$\frac{\psi(x^\alpha, L(x)^\beta)}{x^\alpha} = L(x)^{-\frac{\alpha}{2\beta}+o(1)}.$$

Analysis

Prop. The cost of the naive algorithm is $O(L^{2+o(1)})$.
Proof.
Proba($x(e)$ is p_k-smooth)$= \frac{\psi(N, p_k)}{N} \Rightarrow$ we need $k\frac{N}{\psi(N,p_k)}$ relations.
Using trial division, testing p_k-smoothness costs k divisions.
Linear algebra costs $O(k^r)$ with $2 \le r \le 3$ (see later).
Total cost is:

$$O\left(k^2 \frac{N}{\psi(N, p_k)}\right) + O(k^r).$$

Put $k = L(N)^b$, from which $p_k \approx k \log k = O(L(N)^{b+o(1)})$. Cost is now:

$$O(L^{2b} L^{1/(2b)}) + O(L^{rb}) = O(L^{\max(2b+1/(2b), rb)}).$$

$2b + 1/(2b)$ is minimal for $b = 1/2$ and has value 2, which is larger than rb for all r. □

III. CFRAC

Pb. The above method is not practical, since factoring the relations is too costly. Can we build residues of size N^α for $\alpha < 1$?

Morrison and Brillhart: use the continued fraction expansion of $\sqrt{N}$, leads to residues of size $N^{1/2}$; first real-life algorithm, factored F_7 in 1970. Complexity is $O(L(N)^{\sqrt{2}})$.

Thm. We have $\sqrt{N} = [(a_n)] = [a_0, \overline{a_1, a_2, \ldots, a_n, 2a_0}]$ with

$$u_0 = 0, v_0 = 1,$$

$$\alpha_n = a_n' = [a_n, \ldots] = \frac{u_n + \sqrt{N}}{v_n}, a_n = \lfloor \alpha_n \rfloor, \quad \frac{p_n}{q_n} = [a_0, \ldots, a_n]$$

$$u_{n+1} = a_n v_n - u_n, \quad v_{n+1} = \frac{N - u_{n+1}^2}{v_n},$$

$$p_{n-1}^2 - N q_{n-1}^2 = (-1)^n v_n, |v_n| \le 2\sqrt{N}, p_{-2} = 0, p_{-1} = 1$$

CFRAC: numerical example

Generate many identities:
$p_{n-1}^2 \equiv (-1)^n v_n \bmod N$ with $|v_n| \leq 2\sqrt{N}$.

Prop. If $P \mid v_n$, then $\left(\frac{N}{P}\right) = +1$.
Use: $p_{n-1}^2 \equiv N q_{n-1}^2 \bmod P$ and $\gcd(p_{n-1}, q_{n-1}) = 1$.

Ex. $N = 1147$, $B = \{-1, 2, 3, 11\}$

n	p_{n-1}	v_n	fact $(-1)^n v_n$
4	271	33	$3 \cdot 11$
5	508	11	-11
7	1025	27	-3^3
8	395	33	$3 \cdot 11$
11	941	3	3
13	724	3	3

$$L_5 \cdot L_7 \cdot L_8 : (p_5 \cdot p_7 \cdot p_8)^2 \equiv 3^4 \cdot 11^2 \bmod N$$

$$25^2 \equiv (3^2 \cdot 11)^2 \bmod N \Rightarrow \gcd(25 - 99, N) = 37.$$

CFRAC: implementation tricks

Better formulas: for the recurrence relations.

Multiplier: factor kN instead of N for small k so as to have a lot of small prime factors in $\mathcal{B} = \{p, \left(\frac{kN}{p}\right) = +1\}$.

Early Abort Strategy: if, after removing $p \leq 10^3$, the cofactor of v_n is too large, drop it.

Pomerance: let $0 < c, \theta < 1$.

- Trial divide v_n with primes $\leq L^{\theta b}$.
- If $v_n / \prod_{p < L^{\theta b}} p > N^{(1-c)/2}$, throw v_n away.

Optimal values (when $r = 3$): $b = 1/\sqrt{7}, c = 1/7, \theta = 1/2$, leading to $L^{\sqrt{7}/2 + o(1)}$.

IV. Quadratic sieve

Pbs with CFRAC:

- uses multiprecision computations to generate residues;
- parallelization is a challenge (Williams & Wunderlich, 1987).

Schroeppel's linear sieve: relations
$F(a, b) = (\lfloor\sqrt{N}\rfloor + a)(\lfloor\sqrt{N}\rfloor + b) - N$ for small a and b satisfy

$$F(a, b) \equiv (\lfloor\sqrt{N}\rfloor + a)(\lfloor\sqrt{N}\rfloor + b) \bmod N$$

and $N = \lfloor\sqrt{N}\rfloor^2 + R$, $R = O(\sqrt{N}) \Rightarrow F(a, b)$ have size $O(\sqrt{N})$.

Sieving: if $p \mid F(a, b)$, then $p \mid F(a + p, b)$, etc.

The ancester: Eratosthenes

Eratosthenes: how do we enumerate prime numbers in $[2, X]$? It is much easier to enumerate composite numbers and then deduce the primes as being not composite.

	2**2**	3**3**	4	5**5**	6	7**7**	8	9	10
			2		**2**		**2**	**3**	**2**
					3				**5**
11**11**	12	13**13**	14	15	16	17**17**	18	19**19**	20
	2		**2**	**3**	**2**		**2**		**2**
	3		**7**	**5**			**3**		**5**

- empty sets denote primes;
- non-empty sets contain the prime factors of the composite number.
- we can sieve as well with primes powers: 4, 8, etc.

Eratosthenes: the algorithm

1. Set $T[x] = \emptyset$ for $x \in [2, X]$;
2. **for** $p = 2$ **to** $\sqrt{X}$
   2.1 **if** $T[p] \neq \emptyset$ **then continue**;
   2.2 $x := p^2$;
   2.3 **while** $x \leq X$ **do**
      2.3.1 $T[x] := T[x] \cup \{p\}$;
      2.3.2 $x := x + p$;

Postsieve: find all x s.t. $T[x] = \emptyset$.

Rem. replace 2.3.2 by $x := x + 2p$ as soon as $p > 2$. Some tricks are available (Brent, etc.).

Finding smooth numbers using a sieve

Minor variant of Eratosthenes for a prime basis $\mathcal{B}$:

1. Set $T[x] = []$ for $x \in [2, X]$;
2. **for all** $p \in \mathcal{B}$ and $p^e \leq X$ **do**
   2.2 $x := p^e$;
   2.3 **while** $x \leq X$ **do**
      2.3.1 append$(T[x], p)$; {*trick!*}
      2.3.2 $x := x + p^e$;

Postsieve: find all x s.t. $x = \prod_{p \in T[x]} p$.

Cost: (forgetting the p^e for $e > 1$)

$$\sum_{i=1}^{k} \frac{X}{p} \approx X \sum_{i=1}^{k} \frac{1}{p} \approx X \int_2^k \frac{\mathrm{d}\,t}{t \log t} \approx X \log\log k \approx X \log\log p_k$$

Moreover: no division!

Finding smooth numbers using a sieve (2/2)

A numerical variant of Eratosthenes for $\mathcal{B}$:

1. Set $T[x] = []T[x] = -\lg x$ for $x \in [2, X]$;
2. **for all** $p \in \mathcal{B}$ and $p^e \leq X$ **do**
   2.2 $x := p^e$;
   2.3 **while** $x \leq X$ **do**
      2.3.1 append$(T[x], p) T[x] := T[x] + \lg p$;
      2.3.2 $x := x + p^e$;

Postsieve: find all x s.t. $x = \prod_{p \in T[x]} p T[x] \approx 0$ (enough to have $|T[x]| < \log \max \mathcal{B}$).

Rem. Some tuning is necessary in practice and depending on applications.

Trick: test sign bit!

Extension to polynomials

Pb. Given a polynomial $f(X) \in \mathbb{Z}[X]$, find all $\mathcal{B}$-smooth numbers $f(x)$ for $x \in I = [1, X]$.

Principle: Taylor's formula yields

$$f(x + kq) \equiv f(x) + q(\ldots) \equiv f(x) \bmod q$$

Sieving:

for all roots r of $f \bmod q^e$ **do**
  $x := r$;
  while $x \leq X$ **do**
    append$(T[x], q)$;
    $x := x + q^e$.

Rem. No need to evaluate $f(x)$ until post-sieving.

Pomerance's quadratic sieve:

Use $a = b$ in Schroeppel's sieve:

$$\left(a + \left\lfloor\sqrt{N}\right\rfloor\right)^2 \equiv \left(a + \left\lfloor\sqrt{N}\right\rfloor\right)^2 - N \approx 2a\sqrt{N}.$$

$$p \mid F(a) \Longleftrightarrow \left(a + \left\lfloor\sqrt{N}\right\rfloor\right)^2 \equiv N \bmod p$$

$\Rightarrow \left(\frac{N}{p}\right) = 1$ and $p \mid F(a) \Leftrightarrow a \equiv a_-$ or $a \equiv a_+ \bmod p$.

Prop. The cost is $O(L(N)^{r/\sqrt{4(r-1)}})$.

Proof. Precomputing all roots of $F(a) \bmod p$ costs L^b.
Cost of sieving over $|a| \leq L^c$ is

$$\sum_{p \leq L^b} \frac{2L^c}{p} = L^{c+o(1)}.$$

The number of L^b-smooth values of $F(a)$ in the interval is $L^{c-1/(4b)} \Rightarrow$ take $c = b + 1/(4b)$ and optimize $L^{\max(b, b+1/(4b), rb)} \Rightarrow$ $b = 1/\sqrt{4(r-1)}$. $\square$

Large primes

Idea: suppose we end up with

$$x(e)^2 = (\prod p)C$$

for some $p_k < C(e) < p_k^2$. Then we know that $C(e)$ is prime.
Keep the relation and hope for another

$$x(e')^2 = (\prod p)C$$

so that $(x(e)x(e')/C)^2$ is factored over $\mathcal{B}$. Works due to the birthday paradox. Use hashing to store C's.

More than one prime: filtering (highly technical to implement).

Real sieving in QS

Never factor residues, but test

$$\mathcal{R}(a) = -\log|F(a)| + \sum_{\substack{p^e \mid F(a) \\ p \in \mathcal{B}}} \log p^e > -\log p_k$$

and replace $\log|F(a)|$ by $\log|2a\sqrt{N}|$. In practice, fits in a `char`; use integer approximations; ignore small primes.

Large primes: relax $\mathcal{R}(a) > -2\log p_k$, say.

MPQS: (Montgomery, 1985) use families of quadratic polynomials $\Rightarrow$ massive computations become possible: email (A. K. Lenstra & M. S. Manasse, 1990), INTERNET (RSA-129).

Rem. a lot more tricks exist (SIQS, etc.).