Integer factoring

- The general algorithms to factor n are slow! Currently, only integers in ≈ 700 bits or ≈ 200 digits could be factored; that too using specialized hardware.

- The best provable complexity known is: Expected time $\exp(O(\sqrt{\lg n \cdot \lg\lg n}))$.

- Heuristically, it is $\exp(O(\lg^{1/3} n \cdot \lg^{2/3} \lg n))$.

- We will use the notation $L_x(\alpha, c) :=$
$$\exp(c \cdot \log^{\alpha} x \cdot \log^{1-\alpha} \log x).$$

[Pomerance, 1989]: The general number field sieve (GNFS) has conjectured complexity
$$L_n(1/3, 2).$$

- Why this 'strange' function $L_x(\alpha, c)$?

Smooth numbers

Defn: • A number n is *y-smooth* if all the prime factors of m are $\leq y$.

• Their *density* is denoted by $\psi(x,y)$ $:= \#\{1 < m \leq x \mid n \text{ is } y\text{-smooth}\}$.

– Asymptotic estimate for $\psi(x,y)$ determines the complexity of advanced factoring algorithms.

Thm. (Dickman–de Bruijn '51) $\psi(x,y) \geqslant x/u^u$, $u := \log_y x$.

Pf idea (for a weaker bound)

• Consider the regime $\log y < u < t := y/\log y$.

• There are roughly t primes $2 = p_1 < p_2 < \dots < p_t$ below y.

• Any good m can be expressed as $\prod_{i=1}^{t} p_i^{\alpha_i}$.

• Clearly, $\psi(x,y) \geqslant \#\{\bar{\alpha} \mid \sum_{i=1}^{t} \alpha_i \leq \log_y x\}$

$$\geqslant \binom{u+t}{u} \geqslant \left(\frac{t}{u}\right)^u \geqslant \frac{y^u}{(\log y)^u \cdot u^u} = \frac{x}{(\log x)^u} . \qquad \square$$

- This bound is sensible only when $u^u \ll x$.
Thus, y should not be too small.

▷ A useful, tolerable y is $L_x(\alpha, c)$, for constants α, c, with $u^u \approx L_x\left(1-\alpha, \frac{1-\alpha}{c}\right)$.

Pf idea:

- $\log y \approx \log^{\alpha} x \cdot \log^{1-\alpha} \log x$

$$\Rightarrow u = \frac{\log x}{\log y} \approx \frac{1}{c} \cdot \log^{1-\alpha} x \cdot \log^{\alpha-1} \log x$$

$$\Rightarrow u \cdot \log u \approx \frac{1}{c}\left(\log^{1-\alpha} x \cdot \log^{\alpha-1} \log x\right) \cdot (1-\alpha) \cdot \log\log x$$

$$\approx \frac{1-\alpha}{c} \cdot \log^{1-\alpha} x \cdot \log^{\alpha} \log x.$$

$$\Rightarrow u^u \approx L_x\left(1-\alpha, \frac{1-\alpha}{c}\right). \qquad \square$$

▷ Thus, for $y = L_x(\alpha, c)$ the probability of choosing a y-smooth $m \le x$ is given by

$$\frac{\Psi(x,y)}{x} \approx L_x\left(1-\alpha, -\frac{1-\alpha}{c}\right).$$

- In the factoring algorithms, the time spent depends on the bound y & the probability bound.

- A time complexity of $L_n(\alpha, c)$, $\alpha < 1$, is termed "<u>subexponential</u>", in contrast to the exponential $L_n(1, c)$.

  Eg. Eratosthenes sieve takes $L_n(1, \frac{1}{2})$ time.

Factoring algorithms

- We will start with algorithms that are better than the brute-force on <u>certain</u> n.

Pollard's rho method (1975)

- <u>Idea</u> is to exploit the presence of a "small" prime factor $p \mid n$.

Input: odd $n > 1$ & a pseudorandom function $f(x)$ (say, $f(x) = x^2 + 1 \mod n$).

Output: Factors n heuristically in $\tilde{O}(\sqrt{p} \cdot \lg n)$ time.

1) Randomly pick x ; $y = x$; $d = 1$.
2) While $d = 1$
   - $x = f(x)$; $y = f \circ f(y)$.
   - $d = \gcd(x - y, n)$.
3) If $d \neq n$ then OUTPUT d, else FAIL.

Assumption: p is the smallest prime factor of n & $\{ f^i(x) \mid i \geq 0 \}$ is a random sequence.

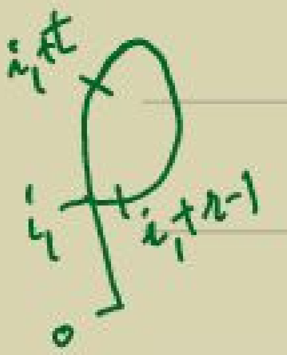
Lemma: Whp $p \mid (x-y)$ after $O(\sqrt{p})$ iterations.

Pf:

- Consider the sequence $\{ f^i(x) \mod p \mid 0 \leq i \leq j \}$.
- The probability of them being distinct $\leq \frac{p}{p} \cdot \frac{p-1}{p} \cdots \frac{p-j}{p} \approx e^{-\left(\frac{1}{p} + \frac{2}{p} + \cdots + \frac{j}{p}\right)} \approx e^{-j^2/p}$.

$\Rightarrow$ For $j = O(\sqrt{p})$, the probability of a repetition is good. (Birthday paradox)

- Thus, $\exists\ 0 \le i_1 < i_2$, $i_2 - i_1 = O(\sqrt{p})$ s.t. $f^{i_1}(x) \equiv f^{i_2}(x) \pmod{p}$.



$\Rightarrow$ After the i_1-th iteration the period $r \le i_2 - i_1 = O(\sqrt{p})$.

$\Rightarrow$ At the (i_1+t)-th iteration we get a <u>collision</u> if $(i_1+t) \equiv 2(i_1+t) \bmod r$,

$\Rightarrow$ $t = r - i_1$ works

$\Rightarrow$ $p \mid x-y$ at the $O(\sqrt{p})$-th iteration whp.

$\square$

- Thus, whp $p \mid d$ in $\tilde{O}(\sqrt{p}.\lg n)$ time.

- Heuristically, we can say that this d will factor n.

<u>Success</u>: Brent & Pollard (1980) factored Fermat number $F_8 = 2^{2^8}+1$ into primes of

16 & 62 digits in 2 hours on a UNIVAC.

Pollard's $p-1$ method (1974)

- It exploits the smoothness of $(p-1)$, for a prime factor $p | n$.

Input: odd $n>1$ not a perfect power.
Output: Factors n.

1) For $r = 2, 3, 4, \ldots$
   - Randomly pick $a \in (\mathbb{Z}/n\mathbb{Z})^*$
   - $d = (a^k - 1, n)$, $k = (r!)^{\lceil \lg n \rceil}$.
   - If $d \notin \{1, n\}$, then OUTPUT d.

Assumption: $\exists$ distinct primes $p, q | n$ s.t. $(p-1)$ is R-smooth but $(q-1)$ is not.

Lemma: Whp n is factored in $\tilde{O}(R \cdot \lg^2 n)$ time.

Proof: • When r reaches R: $\forall a \in (\mathbb{Z}/n\mathbb{Z})^*$,
$a^k \equiv 1 \pmod{p}$ $[\because (p-1) \mid k]$.

• But, for $< \frac{1}{2}$ of the $a \in (\mathbb{Z}/n\mathbb{Z})^*$,
$a^k \not\equiv 1 \pmod{q}$.

$\Rightarrow$ Whp $p \mid (a^k - 1)$ & $q \nmid (a^k - 1)$.

• Time to compute $a^k - 1 \pmod{n}$ is:
$\tilde{O}(\lg k \cdot \lg n) = \tilde{O}(\lg n \cdot r \cdot \lg n)$.
$\Rightarrow$ Time overall (doing binary search $\leq R$)
$= \tilde{O}(R \cdot \lg^2 n)$. $\square$

▷ If we run this for a single (large enough) r, then the complexity is $\tilde{O}(r \cdot \lg^2 n)$.

Success: In GIMPS (Great Internet Mersenne Prime Search), this is used to eliminate composites.

▷ In RSA, $(p-1)$ & $(q-1)$ should not be smooth!

Fermat's method

- Tries to write $n = a^2 - b^2$.
  Works well when a factor of n is very close to $\sqrt{n}$.

▷ $n = cd = \left(\frac{c+d}{2}\right)^2 - \left(\frac{c-d}{2}\right)^2$.
both are odd.

▷ $c \approx \sqrt{n} \iff d \approx \sqrt{n} \iff (c-d)$ is "small".

- So, we can find $\frac{c-d}{2}$ by brute-force.

Algo: 1) For $x = 1, 2, 3, \ldots$
- If $(n+x^2)$ is square, then compute $y = \sqrt{n+x^2}$ & OUTPUT $(y-x)$.

▷ It has time complexity $\tilde{O}(m \cdot \lg n)$, where $m := \min\{c-d \mid n = c \cdot d\}$.

- (Lehman '74) made this general purpose with time complexity $\tilde{O}(n^{1/3})$.

<u>Success</u>: The fundamental idea of finding two <u>squares</u>, congruent modulo n, appears in many advanced algorithms.

Also, known as <u>Kraitchik's family</u> of algorithms. Starting idea from 1920s:

- Consider $Q(x) := X^2 - n$.
- Find $x_1, \dots, x_k$ s.t. $Q(x_1) \cdots Q(x_k)$ is a square v^2.

$\Rightarrow (x_1 \cdots x_k)^2 \equiv v^2 \pmod{n}$.

$\Rightarrow \gcd(x_1 \cdots x_k - v, n)$ might factor n!

<u>Modification</u> by <u>Lehmer</u> & <u>Powers</u> (1931).

- Compute the <u>continued fraction</u> (say, for k' terms): $\sqrt{n} = a_0 + \frac{1}{a_1 +} \frac{1}{a_2 +} \frac{1}{a_3 +} \cdots$

- It is known that the corresponding convergents $\{a_0, x_1/y_1, x_2/y_2, \ldots\}$ give improving approx. to $\sqrt{n}$ & satisfy:

$$Q_i := x_i^2 - n y_i^2 \ , \quad |Q_i| < 2\sqrt{n}.$$

- Since Q_i's are "small", one hopes to quickly find $i_1 < i_2 < \ldots < i_k$ s.t. $Q_{i_1} \cdots Q_{i_k}$ is a square v^2.

$$\Rightarrow \quad Q_{i_1} \cdots Q_{i_k} \equiv (x_{i_1} \cdots x_{i_k})^2 \equiv v^2 \pmod{n}.$$

<u>Morrison & Brillhart's implementation</u> (1970)

- Idea is to use Q_i's, from the continued fraction of $\sqrt{n}$, that are <u>B-smooth</u>.

<u>Input</u>: non-square $n \in \mathbb{N}_{>2}$.
<u>Output</u>: Factoring n.

1) Fix a bound B. Let $\{p_1, \ldots, p_B\}$ be the first B prime numbers. <u>factor-base</u>

2) Compute the set $S := \{\overset{x_i^2 - n\cdot y_i^2}{Q_i} \mid Q_i = (-1)^{\alpha_{i0}} p_1^{\alpha_{i1}} \cdots p_B^{\alpha_{iB}}$, for some $\alpha_{ij} \geq 0\}$ s.t. $|S| = B+2$.

3) Consider the $B+2$ vectors $\{\overset{=:\overline{\alpha_i}}{(\alpha_{i0}, \dots, \alpha_{iB})} \mid Q_i \in S\}$.
Compute a subset $T \subseteq S$ s.t. the vectors $\{\overline{\alpha_i} \mid Q_i \in T\}$ sum to $0 \pmod 2$.

4) So, we have: $\prod_{Q_i \in T} Q_i$ is a square v^2.

5) OUTPUT $\gcd\left(\prod_{Q_i \in T} x_i - v, n\right)$.

<u>Assumption:</u> $\{Q_i = x_i^2 - n\cdot y_i^2 \mid i > 0\}$ is a random sequence.

<u>Theorem</u>: Heuristically, the algorithm takes time
$\exp\left((o(1) + \sqrt{2}) \cdot \sqrt{\lg n \cdot \lg\lg n}\right) = L_n\left(\frac{1}{2}, \sqrt{2} + o(1)\right)$.

<u>Proof</u>:

- $\Pr[Q_i \text{ is } p_B\text{-smooth}] \approx \psi(\sqrt{n}, p_B)/\sqrt{n}$
- $\Rightarrow$ The expected # i's after which we

will get $(B+2)$ smooth Q_i's
$$\approx B \cdot \sqrt{n} / \psi(\sqrt{n}, p_B)$$

- Total time is dominated by the p_B-smoothness check, which requires $\approx B^2 \cdot \sqrt{n}/\psi(\sqrt{n}, p_B)$ time.

- Setting $B = L_{\sqrt{n}}(\alpha, c)$ this becomes
$$\approx L_{\sqrt{n}}(\alpha, c)^2 \cdot L_{\sqrt{n}}\left(1-\alpha, \frac{1-\alpha}{c}\right)$$
$$= e^{2 \cdot c \cdot (\log \sqrt{n})^{\alpha} \cdot (\log\log \sqrt{n})^{1-\alpha} + \frac{1-\alpha}{c} \cdot (\log \sqrt{n})^{1-\alpha} \cdot (\log\log \sqrt{n})^{\alpha}}$$

- which is minimized, at $\alpha = c = \frac{1}{2}$, to:
$$\approx \exp\left(\sqrt{2 \cdot \log n \cdot \log\log n}\right) = L_n\left(\tfrac{1}{2}, \sqrt{2}\right).$$

- $B \approx \exp\left(\frac{1}{2\sqrt{2}} \cdot \sqrt{\log n \cdot \log\log n}\right) = L_n\left(\frac{1}{2}, \frac{1}{2\sqrt{2}}\right)$. □

<u>Success</u>: $F_7 = 2^{128}+1$ was factored into two primes (17 & 22 digits), amongst other ≤ 70 digit numbers.

C.frac. of $\sqrt{257 F_7}$ was used.

Quadratic Sieve

- Pomerance (1981) suggested a sieving idea to reduce the time taken to test smoothness.
  Also, the C. frac. method is too "sequential". It was replaced by $Q(x) = x^2 - n$ where x is kept close to $\sqrt{n}$.

Modifications:

1) In the above algorithm compute the list of $Q(x) = x^2 - n$ for $N := B \cdot \sqrt{n} / \psi(\sqrt{n}, p_B)$ x's above $\lfloor \sqrt{n} \rfloor$.

2) Check their smoothness as:

   For $1 \le i \le B$:

   Why? look at the recurrence.

   Look at $2N/p_i$ places in the list that are divisible by p_i. Modify the list by dividing these by the highest power of p_i.

3) The places in the list, with value $= 1$,

indicate the i's where $\{Q(\lfloor\sqrt{n}\rfloor+i)\}$ is p_B-smooth.

- Time taken now $\approx \sum_{i=1}^{B} \frac{2N}{p_i} \approx N \cdot \log\log B$
$\approx B \cdot \log\log B \cdot \sqrt{n}/\psi(\sqrt{n}, p_B)$
$\approx L_{\sqrt{n}}(\alpha, c) \cdot L_{\sqrt{n}}\left(1-\alpha, \frac{1-\alpha}{c}\right)$, for $B = L_{\sqrt{n}}(\alpha, c)$.

- This gets minimized, at $(\alpha, c) = \left(\frac{1}{2}, \frac{1}{\sqrt{2}}\right)$, to:
$\approx L_{\sqrt{n}}\left(\frac{1}{2}, \frac{1}{\sqrt{2}}\right) \cdot L_{\sqrt{n}}\left(\frac{1}{2}, \frac{1}{\sqrt{2}}\right)$
$\approx L_n\left(\frac{1}{2}, 1\right)$,
for $B \approx L_n\left(\frac{1}{2}, \frac{1}{2}\right)$.

- The drop of $\sqrt{2}$ from the exponent leads to a <u>two-fold</u> increase in the length of n that can be factored!

<u>Success</u>: Lenstra & Manasse (1994) factored a 129-digit RSA challenge using distributed computing over the Internet.

Number field sieve (NFS)

- Pollard (1988) suggested using algebraic number fields to factor numbers of the form x^3+k, for small k & large x.

Lenstra, Lenstra & Manasse (1990) improved it & factored $F_9 = 2^{512}+1$ into 3 primes (7, 49, 99 digits).

Idea: • Quadratic sieve devises equalities of the form $\prod_{i=1}^{k}(x_i^2 - ny_i^2) = v^2$ over $\mathbb{Z}$.

• Using the norm $N : \mathbb{Q}(\sqrt{n}) \to \mathbb{Q}$ we can rewrite it as: $\prod_i N(x_i - y_i\sqrt{n}) = N\left(\prod_i (x_i - y_i\sqrt{n})\right) = v^2$.

• Its high-order generalization is to go to a number field $K = \mathbb{Q}(\alpha) = \mathbb{Q}[x]/\langle f \rangle$, where

$f(m) \equiv_n 0$

f is an irreducible polynomial of degree $(d+1)$. [Use the "smooth numbers" in $\mathbb{Z}[\alpha]$.]

- The $\underline{\text{norm}}$ to consider is $N : \mathbb{Z}[\alpha] \to \mathbb{Z}$ mapping $a_0 + a_1\alpha + \dots + a_d\alpha^d \mapsto \prod_{\beta \in Z(f) \cap \mathbb{C}} (a_0 + a_1\beta + \dots + a_d\beta^d)$

- Eg. $\mathbb{Q}(\sqrt[3]{2}) = \mathbb{Q}[x] / \langle x^3 - 2 \rangle =: \mathbb{Q}(\alpha)$, where α is of deg $= 3$.

  The norm maps $a_0 + a_1 2^{1/3} + a_2 \cdot 2^{2/3}$

  $\mapsto (a_0 + a_1 \cdot \alpha + a_2 \alpha^2) \cdot (a_0 + a_1 \alpha w + a_2 \alpha^2 w^2) \cdot (a_0 + a_1 \alpha w^2 + a_2 \alpha^2 w)$

  where $w = \sqrt[3]{1} \in \mathbb{C}$.

- Hope to find two squares in $\mathbb{Z}[\alpha]$ that are "congruent" mod n.

Input: Large n.
Output: Factoring n.
Algo:

1) Fix a degree d, $m = \lfloor n^{1/d} \rfloor$.

   Express n in base m, say $n = m^d + c_{d-1} m^{d-1} + \dots + c_1 m + c_0$. Consider $f(x) := x^d + c_{d-1} x^{d-1} + \dots + c_1 x + c_0 \in \mathbb{Z}[x]$.

2) Factor $f(x)$ by L^3.

If it factors then we factor n or pick an irred. factor as f.

3) Now f is irreducible: Consider the number field $\mathbb{Q}[X]/\langle f(X)\rangle =: \mathbb{Q}(\alpha)$.

- $[\mathbb{Q}(\alpha):\mathbb{Q}] = d$.
- We have a homomorphism φ from the "integers"

$$\varphi: \begin{array}{ccc} \mathbb{Z}[\alpha] & \longrightarrow & \mathbb{Z}/n\mathbb{Z}, \\ \alpha & \longmapsto & m \end{array}$$

- We have a norm in $\mathbb{Q}(\alpha)$,

$$N: \mathbb{Q}(\alpha) \longrightarrow \mathbb{Q}$$

$$a_0 + \dots + a_{d-1}\alpha^{d-1} \longmapsto \prod_{\beta\in\mathbb{C},\, f(\beta)=0} (a_0 + a_1\beta + \dots + a_{d-1}\beta^{d-1})$$

4) Sieving: For a carefully chosen (u, y), find $U \subseteq \{(a,b)\in\mathbb{Z}^2 \mid a, b \le u\}$ s.t. both $(a-bm)$ & $N(a-b\alpha)$ are y-smooth.

(for a sq. in $\mathbb{Z}$) (for a sq. in $\mathbb{Z}[\alpha]$)

$[\triangleright\ N(a-b\alpha) = b^d \cdot f(\frac{a}{b}) = \sum_{i=0}^{d} c_i a^i b^{d-i}.\ ]$

[▷ $a - b\alpha$ factors, in the ring of integers O_K of K, into <u>prime ideals</u> $\mathfrak{q}_i \triangleleft O_K$.
▷ Further, $N(a-b\alpha) = \prod_i q_i^{e_i}$ if $\mathfrak{q}_i \triangleleft \mathbb{Z}[\alpha]$, in which case each $\mathfrak{q}_i$ corresponds to a prime q_i.
In fact, every prime ideal $\mathfrak{q} \mid (a-b\alpha)$ is in <u>1-1 correspondence</u> with a prime q & $r \in \mathbb{F}_q$ s.t. $a - br = f(r) = 0$ in $\mathbb{F}_q$.]

5) <u>Matrix reduction</u>: Find $U' \subseteq U$ s.t.

- $\prod_{a,b \in U'} (a-bm) = v^2$ in $\mathbb{Z}$, &

- $\prod_{a,b \in U'} (a-b\alpha) = \gamma^2$ in $\mathbb{Z}[\alpha]$.

6) OUTPUT $\gcd(v - \phi(\gamma), n)$.

<u>Analysis</u>:

- NFS needs all the algorithms that we have seen in the course — fast integer/matrix mult., polynomial fact. over $\mathbb{F}_p$ or $\mathbb{Q}$, gcd, primality!

- The time complexity of NFS is dominated by $\underset{\text{Sieving}}{u^{2+o(1)}} + \underset{\text{Matrix reduction}}{y^{2+o(1)}}$.

- So, we intend $\log u \approx \log y$.

- The integer $(a-bm)\cdot N(a-b\alpha) \approx (a-bm)\cdot \sum_{0\le i\le d} c_i a^i b^{d-i} \approx un^{1/d}\cdot n^{1/d}\cdot u^d \approx u^{d+1}\cdot n^{2/d}$.

▷ A number $\le u^{d+1}\cdot n^{2/d}$ is y-smooth with probability r^{-r}, where $r = \log_y(u^{d+1} n^{2/d}) \approx \log(u^{d+1} n^{2/d})/\log u$.

- To maximize the probability we minimize $r = d+1 + (2/d)\cdot \log_u n$.

$\Rightarrow d \approx \sqrt{2\log_u n} \Rightarrow r \approx 2\cdot\sqrt{2\log_u n}$.

- To get the squares, via matrix reduction, we need $\#U \approx y \Rightarrow u^2\cdot r^{-r} \approx y$

$\Rightarrow \log u \approx r\log r \Rightarrow r \approx \log u/\log\log u$.

$\Rightarrow \sqrt{8\log_u n} \approx \log u / \log\log u$

$\Rightarrow 2\cdot(\log n)^{1/3} \approx (\log u)\cdot(\log\log u)^{-2/3}$

$\Rightarrow \log u \approx 2\cdot(\log n)^{1/3}\cdot(\log\log u)^{2/3}$

$\approx 2\cdot(\log n)^{1/3}\cdot(\frac{1}{3}\cdot\log\log n)^{2/3}$

$\Rightarrow y \approx u \approx L_n(1/3, \sqrt[3]{8/9})$.

▷ The time complexity is $L_n(\frac{1}{3}, \sqrt[3]{64/9})$.
The degree $d = (3\log n/\log\log n)^{1/3}$.

<u>The algebraic obstructions</u>

(i) $\mathbb{Z}[\alpha]$ is possibly <u>not</u> $\mathcal{O}_K$.
Thus, γ may not exist in $\mathbb{Z}[\alpha]$, and then $\phi(\gamma)$ does not make sense.
▷ $\forall a \in \mathcal{O}_K$, $f'(\alpha)\cdot a \in \mathbb{Z}[\alpha]$.

(ii) For $a \in \mathcal{O}_K$, the exponent of a wrt every prime $\mathfrak{q} \triangleleft \mathbb{Z}[\alpha]$ may be <u>even</u>, <u>without</u> a being a <u>square</u>.

▷ If $a-b\alpha \pmod{P}$ is a <u>square</u> for <u>random</u> $O(\lg n)$ primes $P \triangleleft \mathbb{Z}[\alpha]$, then whp $(a-b\alpha)$ is a square in O_K (up to a unit).

(iii) O_K has <u>infinitely</u> many <u>units</u> unlike $\mathbb{Z}$.
We need to identify them.

▷ $|O_K^* / O_K^{*2}| \le 2^d$.

- Thus, algebraic number theory takes care of all the obstructions (heuristically).

<u>Success</u>: Largest numbers factored are by NFS.
e.g. $2^{1157}+1$ (347-digit number).