# Integer factorization

D. J. Bernstein

Thanks to:

University of Illinois at Chicago

NSF DMS–0140542

Alfred P. Sloan Foundation

## Sieving $c$ and $611 + c$ for small $c$:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | 2 | | | | | | | |
| 3 | | | | | 3 | | | |
| 4 | 2 | 2 | | | | | | |
| 5 | | | | | | | 5 | |
| 6 | 2 | | | | 3 | | | |
| 7 | | | | | | | | 7 |
| 8 | 2 | 2 | 2 | | | | | |
| 9 | | | | | 3 | 3 | | |
| 10 | 2 | | | | | | 5 | |
| 11 | | | | | | | | |
| 12 | 2 | 2 | | | 3 | | | |
| 13 | | | | | | | | |
| 14 | 2 | | | | | | | 7 |
| 15 | | | | | 3 | | 5 | |
| 16 | 2 | 2 | 2 | 2 | | | | |
| 17 | | | | | | | | |
| 18 | 2 | | | | 3 | 3 | | |
| 19 | | | | | | | | |
| 20 | 2 | 2 | | | | | 5 | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 612 | 2 | 2 | | | 3 | 3 | | | | | | |
| 613 | | | | | | | | | | | | |
| 614 | 2 | | | | | | | | | | | |
| 615 | | | | | 3 | | | 5 | | | | |
| 616 | 2 | 2 | 2 | | | | | | | | | 7 |
| 617 | | | | | | | | | | | | |
| 618 | 2 | | | | 3 | | | | | | | |
| 619 | | | | | | | | | | | | |
| 620 | 2 | 2 | | | | | | 5 | | | | |
| 621 | | | | | 3 | 3 | 3 | | | | | |
| 622 | 2 | | | | | | | | | | | |
| 623 | | | | | | | | | | | | 7 |
| 624 | 2 | 2 | 2 | 2 | 3 | | | | | | | |
| 625 | | | | | | | | 5 | 5 | 5 | 5 | |
| 626 | 2 | | | | | | | | | | | |
| 627 | | | | | 3 | | | | | | | |
| 628 | 2 | 2 | | | | | | | | | | |
| 629 | | | | | | | | | | | | |
| 630 | 2 | | | | 3 | 3 | | 5 | | | | 7 |
| 631 | | | | | | | | | | | | |

etc.

on

is at Chicago

42

oundation

Sieving $c$ and $611 + c$ for small $c$:

| $c$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | 2 | | | | | | | |
| 3 | | | | | 3 | | | |
| 4 | 2 | 2 | | | | | | |
| 5 | | | | | | | 5 | |
| 6 | 2 | | | | 3 | | | |
| 7 | | | | | | | | 7 |
| 8 | 2 | 2 | 2 | | | | | |
| 9 | | | | | 3 | 3 | | |
| 10 | 2 | | | | | | 5 | |
| 11 | | | | | | | | |
| 12 | 2 | 2 | | | 3 | | | |
| 13 | | | | | | | | |
| 14 | 2 | | | | | | | 7 |
| 15 | | | | | 3 | | 5 | |
| 16 | 2 | 2 | 2 | 2 | | | | |
| 17 | | | | | | | | |
| 18 | 2 | | | | 3 | 3 | | |
| 19 | | | | | | | | |
| 20 | 2 | 2 | | | | | 5 | |

| $611+c$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 612 | 2 | 2 | | | 3 | 3 | | | | | | |
| 613 | | | | | | | | | | | | |
| 614 | 2 | | | | | | | | | | | |
| 615 | | | | | 3 | | | 5 | | | | |
| 616 | 2 | 2 | 2 | | | | | | | | | 7 |
| 617 | | | | | | | | | | | | |
| 618 | 2 | | | | 3 | | | | | | | |
| 619 | | | | | | | | | | | | |
| 620 | 2 | 2 | | | | | | 5 | | | | |
| 621 | | | | | 3 | 3 | 3 | | | | | |
| 622 | 2 | | | | | | | | | | | |
| 623 | | | | | | | | | | | | 7 |
| 624 | 2 | 2 | 2 | 2 | 3 | | | | | | | |
| 625 | | | | | | | | 5 | 5 | 5 | 5 | |
| 626 | 2 | | | | | | | | | | | |
| 627 | | | | | 3 | | | | | | | |
| 628 | 2 | 2 | | | | | | | | | | |
| 629 | | | | | | | | | | | | |
| 630 | 2 | | | | 3 | 3 | | 5 | | | | 7 |
| 631 | | | | | | | | | | | | |

etc.

Have complete fac

$c(611 + c)$ for som

$14 \cdot 625 = 2^1 3^0 5^4 7$

$64 \cdot 675 = 2^6 3^3 5^2 7$

$75 \cdot 686 = 2^1 3^1 5^2 7$

$14 \cdot 64 \cdot 75 \cdot 625 \cdot 6$

$= 2^8 3^4 5^8 7^4 = (2^4$

gcd$\{14 \cdot 64 \cdot 75 -$

$= 47.$

$611 = 47 \cdot 13.$

Sieving $c$ and $611 + c$ for small $c$:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | 2 | | | | | | | |
| 3 | | | | | 3 | | | |
| 4 | 2 | 2 | | | | | | |
| 5 | | | | | | | 5 | |
| 6 | 2 | | | | 3 | | | |
| 7 | | | | | | | | 7 |
| 8 | 2 | 2 | 2 | | | | | |
| 9 | | | | | 3 | 3 | | |
| 10 | 2 | | | | | | 5 | |
| 11 | | | | | | | | |
| 12 | 2 | 2 | | | 3 | | | |
| 13 | | | | | | | | |
| 14 | 2 | | | | | | | 7 |
| 15 | | | | | 3 | | 5 | |
| 16 | 2 | 2 | 2 | 2 | | | | |
| 17 | | | | | | | | |
| 18 | 2 | | | | 3 | 3 | | |
| 19 | | | | | | | | |
| 20 | 2 | 2 | | | | | 5 | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 612 | 2 | 2 | | | 3 | 3 | | | | | | |
| 613 | | | | | | | | | | | | |
| 614 | 2 | | | | | | | | | | | |
| 615 | | | | | 3 | | | 5 | | | | |
| 616 | 2 | 2 | 2 | | | | | | | | | 7 |
| 617 | | | | | | | | | | | | |
| 618 | 2 | | | | 3 | | | | | | | |
| 619 | | | | | | | | | | | | |
| 620 | 2 | 2 | | | | | | 5 | | | | |
| 621 | | | | | 3 | 3 | 3 | | | | | |
| 622 | 2 | | | | | | | | | | | |
| 623 | | | | | | | | | | | | 7 |
| 624 | 2 | 2 | 2 | 2 | 3 | | | | | | | |
| 625 | | | | | | | | 5 | 5 | 5 | 5 | |
| 626 | 2 | | | | | | | | | | | |
| 627 | | | | | 3 | | | | | | | |
| 628 | 2 | 2 | | | | | | | | | | |
| 629 | | | | | | | | | | | | |
| 630 | 2 | | | | 3 | 3 | | 5 | | | | 7 |
| 631 | | | | | | | | | | | | |

etc.

Have complete factorization of $c(611 + c)$ for some $c$'s.

$14 \cdot 625 = 2^1 3^0 5^4 7^1$.
$64 \cdot 675 = 2^6 3^3 5^2 7^0$.
$75 \cdot 686 = 2^1 3^1 5^2 7^3$.

$14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$
$= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2$.

$\gcd\{14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2, 611\}$
$= 47$.

$611 = 47 \cdot 13$.

$+\, c$ for small $c$:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | | | 3 | 3 | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | 3 | | | 5 | | | | |
| 2 | 2 | | | | | | | | | 7 |
| | | | | | | | | | | |
| | | | 3 | | | | | | | |
| | | | | | | | | | | |
| 2 | | | | | | 5 | | | | |
| | | | 3 | 3 | 3 | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | 7 |
| 2 | 2 | 2 | 3 | | | | | | | |
| | | | | | | 5 | 5 | 5 | 5 | |
| | | | | | | | | | | |
| | | | 3 | | | | | | | |
| 2 | | | | | | | | | | |
| | | | | | | | | | | |
| | | | 3 | 3 | | 5 | | | | 7 |

Have complete factorization of $c(611 + c)$ for some $c$'s.

$14 \cdot 625 = 2^1 3^0 5^4 7^1$.
$64 \cdot 675 = 2^6 3^3 5^2 7^0$.
$75 \cdot 686 = 2^1 3^1 5^2 7^3$.

$14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$
$= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2$.

$\gcd\{14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2, 611\}$
$= 47$.

$611 = 47 \cdot 13$.

Given $n$ and paran

1. Use powers of p

sieve $c$ and $n + c$

2. Look for nonem

with $c(n + c)$ com

and with $\prod_c c(n +$

3. Compute gcd$\{x$

where $x = \prod_c c -$

Have complete factorization of $c(611+c)$ for some $c$'s.

$14 \cdot 625 = 2^1 3^0 5^4 7^1$.
$64 \cdot 675 = 2^6 3^3 5^2 7^0$.
$75 \cdot 686 = 2^1 3^1 5^2 7^3$.

$14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$
$= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2$.

$\gcd\left\{14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2, 611\right\}$
$= 47$.

$611 = 47 \cdot 13$.

Given $n$ and parameter $y$:

1. Use powers of primes $\le y$ to sieve $c$ and $n+c$ for $1 \le c \le y^2$.

2. Look for nonempty set of $c$'s with $c(n+c)$ completely factored and with $\prod_c c(n+c)$ square.

3. Compute $\gcd\{x, n\}$ where $x = \prod_c c - \sqrt{\prod_c c(n+c)}$.

ctorization of
ne $c$'s.

$7^1$.

$7^0$.

$7^3$.

$675 \cdot 686$

$3^2 5^4 7^2)^2$.

$2^4 3^2 5^4 7^2, 611\}$

Given $n$ and parameter $y$:

1. Use powers of primes $\le y$ to sieve $c$ and $n+c$ for $1 \le c \le y^2$.

2. Look for nonempty set of $c$'s with $c(n+c)$ completely factored and with $\prod_c c(n+c)$ square.

3. Compute $\gcd\{x, n\}$ where $x = \prod_c c - \sqrt{\prod_c c(n+c)}$.

This is the **Q siev**

Same principles:

**Continued-fractio**

(Lehmer, Powers,

Brillhart, Morrison

**Linear sieve** (Sch

**Quadratic sieve** (

**Number-field siev**

(Pollard, Buhler, L

Pomerance, Adlem

Given $n$ and parameter $y$:

1. Use powers of primes $\le y$ to sieve $c$ and $n+c$ for $1 \le c \le y^2$.

2. Look for nonempty set of $c$'s with $c(n+c)$ completely factored and with $\prod_c c(n+c)$ square.

3. Compute $\gcd\{x, n\}$ where $x = \prod_c c - \sqrt{\prod_c c(n+c)}$.

This is the **Q sieve**.

Same principles:

**Continued-fraction method** (Lehmer, Powers, Brillhart, Morrison).

**Linear sieve** (Schroeppel).

**Quadratic sieve** (Pomerance).

**Number-field sieve** (Pollard, Buhler, Lenstra, Pomerance, Adleman).

meter $y$:

primes $\leq y$ to
for $1 \leq c \leq y^2$.

npty set of $c$'s
pletely factored
$c$) square.

$c, n\}$
$\sqrt{\prod_c c(n+c)}$.

This is the **Q sieve**.

Same principles:
**Continued-fraction method**
(Lehmer, Powers,
Brillhart, Morrison).
**Linear sieve** (Schroeppel).
**Quadratic sieve** (Pomerance).
**Number-field sieve**
(Pollard, Buhler, Lenstra,
Pomerance, Adleman).

Sieving speed

Handle sieving in
sieve $\{n+1, \ldots,$
sieve $\{n+y+1,$
sieve $\{n+2y+1,$
etc.

Sieving $\{n+1, n$
using primes $p \leq$
means finding, for
$c \in \{n+1, n+2,$
which $p$'s divide $n$

This is the **Q sieve**.

Same principles:

**Continued-fraction method** (Lehmer, Powers, Brillhart, Morrison).

**Linear sieve** (Schroeppel).

**Quadratic sieve** (Pomerance).

**Number-field sieve** (Pollard, Buhler, Lenstra, Pomerance, Adleman).

<u>Sieving speed</u>

Handle sieving in $y$ pieces:
sieve $\{n+1,\ldots,n+y\}$;
sieve $\{n+y+1,\ldots,n+2y\}$;
sieve $\{n+2y+1,\ldots,n+3y\}$;
etc.

Sieving $\{n+1,n+2,\ldots,n+y\}$ using primes $p \le y$ means finding, for each $c \in \{n+1,n+2,\ldots,n+y\}$, which $p$'s divide $n+c$.

**e**.

**on method**

).

roeppel).

(Pomerance).

**ve**

Lenstra,

nan).

## Sieving speed

Handle sieving in $y$ pieces:
sieve $\{n+1,\ldots,n+y\}$;
sieve $\{n+y+1,\ldots,n+2y\}$;
sieve $\{n+2y+1,\ldots,n+3y\}$;
etc.

Sieving $\{n+1, n+2, \ldots, n+y\}$
using primes $p \le y$
means finding, for each
$c \in \{n+1, n+2, \ldots, n+y\}$,
which $p$'s divide $n+c$.

Consider all pairs (
where $n+c$ is a m

Easy to generate p
sorted by second c
(612, 2), (614, 2),
(620, 2), (612, 3),
(615, 5), (620, 5),

Sieving means listi
sorted by first com
(612, 2), (612, 3),
(615, 3), (615, 5),
(618, 2), (618, 3),

## Sieving speed

Handle sieving in $y$ pieces:
sieve $\{n+1,\ldots,n+y\}$;
sieve $\{n+y+1,\ldots,n+2y\}$;
sieve $\{n+2y+1,\ldots,n+3y\}$;
etc.

Sieving $\{n+1,n+2,\ldots,n+y\}$
using primes $p \le y$
means finding, for each
$c \in \{n+1,n+2,\ldots,n+y\}$,
which $p$'s divide $n+c$.

Consider all pairs $(n+c,p)$
where $n+c$ is a multiple of $p$.

Easy to generate pairs
sorted by second component:
$(612,2)$, $(614,2)$, $(616,2)$, $(618,2)$,
$(620,2)$, $(612,3)$, $(615,3)$, $(618,3)$,
$(615,5)$, $(620,5)$, $(616,7)$.

Sieving means listing pairs
sorted by first component:
$(612,2)$, $(612,3)$, $(614,2)$,
$(615,3)$, $(615,5)$, $(616,2)$, $(616,7)$,
$(618,2)$, $(618,3)$, $(620,2)$, $(620,5)$.

$y$ pieces:

$n + y\}$;

$\ldots, n + 2y\}$;

$, \ldots, n + 3y\}$;

$+ 2, \ldots, n + y\}$

$y$

each

$\ldots, n + y\}$,

$+ c$.

Consider all pairs $(n + c, p)$ where $n + c$ is a multiple of $p$.

Easy to generate pairs sorted by second component: (612, 2), (614, 2), (616, 2), (618, 2), (620, 2), (612, 3), (615, 3), (618, 3), (615, 5), (620, 5), (616, 7).

Sieving means listing pairs sorted by first component: (612, 2), (612, 3), (614, 2), (615, 3), (615, 5), (616, 2), (616, 7), (618, 2), (618, 3), (620, 2), (620, 5).

There are $y^{1+o(1)}$

involving $\{n + 1, n$

Sieving $\{n + 1, n$

takes $y^{1+o(1)}$ seco

on RAM costing $y$

2-dimensional mes

is much faster: $y^0$

on machine costin

Can do even bette

on machine costin

using "elliptic-curv

Consider all pairs $(n+c, p)$ where $n+c$ is a multiple of $p$.

Easy to generate pairs sorted by second component: $(612, 2)$, $(614, 2)$, $(616, 2)$, $(618, 2)$, $(620, 2)$, $(612, 3)$, $(615, 3)$, $(618, 3)$, $(615, 5)$, $(620, 5)$, $(616, 7)$.

Sieving means listing pairs sorted by first component: $(612, 2)$, $(612, 3)$, $(614, 2)$, $(615, 3)$, $(615, 5)$, $(616, 2)$, $(616, 7)$, $(618, 2)$, $(618, 3)$, $(620, 2)$, $(620, 5)$.

There are $y^{1+o(1)}$ pairs involving $\{n+1, n+2, \ldots, n+y\}$.

Sieving $\{n+1, n+2, \ldots, n+y\}$ takes $y^{1+o(1)}$ seconds on RAM costing $y^{1+o(1)}$ dollars.

2-dimensional mesh computer is much faster: $y^{0.5+o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars.

Can do even better: $y^{o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars, using "elliptic-curve method."

$(n + c, p)$
multiple of $p$.

pairs
component:
$(616, 2)$, $(618, 2)$,
$(615, 3)$, $(618, 3)$,
$(616, 7)$.

ing pairs
nponent:
$(614, 2)$,
$(616, 2)$, $(616, 7)$,
$(620, 2)$, $(620, 5)$.

There are $y^{1+o(1)}$ pairs involving $\{n + 1, n + 2, \ldots, n + y\}$.

Sieving $\{n + 1, n + 2, \ldots, n + y\}$ takes $y^{1+o(1)}$ seconds on RAM costing $y^{1+o(1)}$ dollars.

2-dimensional mesh computer is much faster: $y^{0.5+o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars.

Can do even better: $y^{o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars, using "elliptic-curve method."

Square-finding spe

Start from factore
$c_1(n + c_1) = \prod_p p$
$c_2(n + c_2) = \prod_p p$
etc.

Want to find $f_1, f$
such that
$(c_1(n + c_1))^{f_1}(c_2($
is a square.

In other words:
$\prod_p p^{f_1 e_1(p)+f_2 e_2(p}$
has even exponent

There are $y^{1+o(1)}$ pairs involving $\{n+1, n+2, \ldots, n+y\}$.

Sieving $\{n+1, n+2, \ldots, n+y\}$ takes $y^{1+o(1)}$ seconds on RAM costing $y^{1+o(1)}$ dollars.

2-dimensional mesh computer is much faster: $y^{0.5+o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars.

Can do even better: $y^{o(1)}$ seconds on machine costing $y^{1+o(1)}$ dollars, using "elliptic-curve method."

## Square-finding speed

Start from factored $c(n+c)$'s:
$c_1(n+c_1) = \prod_p p^{e_1(p)}$,
$c_2(n+c_2) = \prod_p p^{e_2(p)}$,
etc.

Want to find $f_1, f_2, \ldots$ such that $(c_1(n+c_1))^{f_1}(c_2(n+c_2))^{f_2} \cdots$ is a square.

In other words: $\prod_p p^{f_1 e_1(p) + f_2 e_2(p) + \cdots}$ has even exponents.

pairs

$n+2,\ldots,n+y\}$.

$+2,\ldots,n+y\}$

nds

$^{1+o(1)}$ dollars.

h computer

$^{.5+o(1)}$ seconds

g $y^{1+o(1)}$ dollars.

er: $y^{o(1)}$ seconds

g $y^{1+o(1)}$ dollars,

ve method."

Square-finding speed

Start from factored $c(n+c)$'s:

$c_1(n+c_1) = \prod_p p^{e_1(p)}$,

$c_2(n+c_2) = \prod_p p^{e_2(p)}$,

etc.

Want to find $f_1, f_2, \ldots$

such that

$(c_1(n+c_1))^{f_1}(c_2(n+c_2))^{f_2}\cdots$

is a square.

In other words:

$\prod_p p^{f_1 e_1(p) + f_2 e_2(p) + \cdots}$

has even exponents.

In other words:

$f_1(e_1(2), e_1(3), e_1$

$+ f_2(e_2(2), e_2(3),$

$+ \cdots$ is even.

e.g. given

$14 \cdot 625 = 2^1 3^0 5^4$

$64 \cdot 675 = 2^6 3^3 5^2$

$75 \cdot 686 = 2^1 3^1 5^2$

find $f_1, f_2, f_3$ such

$f_1(1,0,4,1) + f_2($

$+ f_3(1,1,2,3)$ is e

## Square-finding speed

Start from factored $c(n+c)$'s:
$c_1(n+c_1) = \prod_p p^{e_1(p)}$,
$c_2(n+c_2) = \prod_p p^{e_2(p)}$,
etc.

Want to find $f_1, f_2, \ldots$
such that
$(c_1(n+c_1))^{f_1}(c_2(n+c_2))^{f_2}\cdots$
is a square.

In other words:
$\prod_p p^{f_1 e_1(p) + f_2 e_2(p) + \cdots}$
has even exponents.

In other words:
$f_1(e_1(2), e_1(3), e_1(5), \ldots)$
$+ f_2(e_2(2), e_2(3), e_2(5), \ldots)$
$+ \cdots$ is even.

e.g. given
$14 \cdot 625 = 2^1 3^0 5^4 7^1$,
$64 \cdot 675 = 2^6 3^3 5^2 7^0$,
$75 \cdot 686 = 2^1 3^1 5^2 7^3$:
find $f_1, f_2, f_3$ such that
$f_1(1,0,4,1) + f_2(6,3,2,0)$
$+ f_3(1,1,2,3)$ is even.

eed

d $c(n+c)$'s:

$p^{e_1(p)}$,

$p^{e_2(p)}$,

$f_2, \ldots$

$(n+c_2))^{f_2} \ldots$

$) + \cdots$

ts.

In other words:

$f_1(e_1(2), e_1(3), e_1(5), \ldots)$
$+ f_2(e_2(2), e_2(3), e_2(5), \ldots)$
$+ \cdots$ is even.

e.g. given
$14 \cdot 625 = 2^1 3^0 5^4 7^1$,
$64 \cdot 675 = 2^6 3^3 5^2 7^0$,
$75 \cdot 686 = 2^1 3^1 5^2 7^3$:
find $f_1, f_2, f_3$ such that
$f_1(1,0,4,1) + f_2(6,3,2,0)$
$+ f_3(1,1,2,3)$ is even.

This is linear algeb
finding kernel of a

$y^{3+o(1)}$ seconds
using Gaussian eli

$y^{2+o(1)}$ seconds
using Wiedemann'

Again exploit para
$y^{1.5+o(1)}$ seconds
on a 2-dimensiona
costing $y^{1+o(1)}$ do

In other words:

$f_1(e_1(2), e_1(3), e_1(5), \ldots)$
$+ f_2(e_2(2), e_2(3), e_2(5), \ldots)$
$+ \cdots$ is even.

e.g. given
$14 \cdot 625 = 2^1 3^0 5^4 7^1$,
$64 \cdot 675 = 2^6 3^3 5^2 7^0$,
$75 \cdot 686 = 2^1 3^1 5^2 7^3$:
find $f_1, f_2, f_3$ such that
$f_1(1, 0, 4, 1) + f_2(6, 3, 2, 0)$
$+ f_3(1, 1, 2, 3)$ is even.

This is linear algebra mod 2:
finding kernel of a matrix.

$y^{3+o(1)}$ seconds
using Gaussian elimination.

$y^{2+o(1)}$ seconds
using Wiedemann's method.

Again exploit parallelism:
$y^{1.5+o(1)}$ seconds
on a 2-dimensional mesh
costing $y^{1+o(1)}$ dollars.

$(5), \ldots)$

$e_2(5), \ldots)$

$7^1$,

$7^0$,

$7^3$:

n that

$6, 3, 2, 0)$

even.

This is linear algebra mod 2:
finding kernel of a matrix.

$y^{3+o(1)}$ seconds
using Gaussian elimination.

$y^{2+o(1)}$ seconds
using Wiedemann's method.

Again exploit parallelism:
$y^{1.5+o(1)}$ seconds
on a 2-dimensional mesh
costing $y^{1+o(1)}$ dollars.

How big is $y$?

Positive integers $\le$
have $\approx u^{-u}$ chanc
of completely fact
into primes $\le y$,
where $u = (\log x)$

Very crude approx
but in right ballpa

(Try numerical exp
count products of
use fancy analytic

This is linear algebra mod 2: finding kernel of a matrix.

$y^{3+o(1)}$ seconds using Gaussian elimination.

$y^{2+o(1)}$ seconds using Wiedemann's method.

Again exploit parallelism: $y^{1.5+o(1)}$ seconds on a 2-dimensional mesh costing $y^{1+o(1)}$ dollars.

How big is $y$?

Positive integers $\le x$ have $\approx u^{-u}$ chance of completely factoring into primes $\le y$, where $u = (\log x)/\log y$.

Very crude approximation but in right ballpark.

(Try numerical experiments; count products of primes; use fancy analytic theorems.)

ra mod 2:
matrix.

mination.

s method.

llelism:

l mesh
llars.

How big is $y$?

Positive integers $\le x$
have $\approx u^{-u}$ chance
of completely factoring
into primes $\le y$,
where $u = (\log x)/\log y$.

Very crude approximation
but in right ballpark.

(Try numerical experiments;
count products of primes;
use fancy analytic theorems.)

For $\log y \approx \sqrt{(1/2}$
Positive integers $\le$
have $\approx 1/y$ chanc
of completely fact
into primes $\le y$.

Presumably the in
$1(n+1), 2(n+2)$
have $\approx y$ complet
thus produce a sq
often factor $n$.

How big is $y$?

Positive integers $\leq x$
have $\approx u^{-u}$ chance
of completely factoring
into primes $\leq y$,
where $u = (\log x)/\log y$.

Very crude approximation
but in right ballpark.

(Try numerical experiments;
count products of primes;
use fancy analytic theorems.)

For $\log y \approx \sqrt{(1/2)\log n \log\log n}$:
Positive integers $\leq y^2(n + y^2)$
have $\approx 1/y$ chance
of completely factoring
into primes $\leq y$.

Presumably the integers
$1(n+1), 2(n+2), \ldots, y^2(n+y^2)$
have $\approx y$ complete factorizations;
thus produce a square;
often factor $n$.

$\leq x$
ce
oring

$/ \log y$.

imation
rk.

periments;
primes;
theorems.)

For $\log y \approx \sqrt{(1/2) \log n \log \log n}$:
Positive integers $\leq y^2(n + y^2)$
have $\approx 1/y$ chance
of completely factoring
into primes $\leq y$.

Presumably the integers
$1(n+1), 2(n+2), \ldots, y^2(n+y^2)$
have $\approx y$ complete factorizations;
thus produce a square;
often factor $n$.

So we believe that
**Q**-sieve price-perfo
is $c + o(1)$ power o
$\exp(\sqrt{\log n \log \log$
for some constant

Continued-fraction
linear sieve, quadra
smaller power.
Use integers aroun

Number-field sieve
$\exp(\sqrt[3]{(\log n)(\log$
Use even smaller i

For $\log y \approx \sqrt{(1/2)\log n \log\log n}$:
Positive integers $\le y^2(n+y^2)$
have $\approx 1/y$ chance
of completely factoring
into primes $\le y$.

Presumably the integers
$1(n+1), 2(n+2), \ldots, y^2(n+y^2)$
have $\approx y$ complete factorizations;
thus produce a square;
often factor $n$.

So we believe that
$\mathbf{Q}$-sieve price-performance ratio
is $c+o(1)$ power of
$\exp(\sqrt{\log n \log\log n})$,
for some constant $c$.

Continued-fraction method,
linear sieve, quadratic sieve:
smaller power.
Use integers around $\sqrt{n}$.

Number-field sieve: power of
$\exp(\sqrt[3]{(\log n)(\log\log n)^2})$.
Use even smaller integers.

$2) \log n \log\log n$:

$\le y^2(n + y^2)$

e

oring

tegers

$), \ldots, y^2(n + y^2)$

e factorizations;

uare;

So we believe that
$\mathbf{Q}$-sieve price-performance ratio
is $c + o(1)$ power of
$\exp(\sqrt{\log n \log\log n})$,
for some constant $c$.

Continued-fraction method,
linear sieve, quadratic sieve:
smaller power.
Use integers around $\sqrt{n}$.

Number-field sieve: power of
$\exp(\sqrt[3]{(\log n)(\log\log n)^2})$.
Use even smaller integers.

Discrete logarithm

Can use the same
to compute $k$ give
“Index-calculus me

Exponential in log
to compute discret
by collisions, kang

Subexponential in
to use index calcul
Can cryptanalyze

So we believe that
$\mathbf{Q}$-sieve price-performance ratio
is $c + o(1)$ power of
$\exp(\sqrt{\log n \log\log n})$,
for some constant $c$.

Continued-fraction method,
linear sieve, quadratic sieve:
smaller power.
Use integers around $\sqrt{n}$.

Number-field sieve: power of
$\exp(\sqrt[3]{(\log n)(\log\log n)^2})$.
Use even smaller integers.

<u>Discrete logarithms</u>

Can use the same techniques
to compute $k$ given $3^k \bmod n$.
"Index-calculus methods."

Exponential in $\log n$
to compute discrete logarithm
by collisions, kangaroos, etc.

Subexponential in $\log n$
to use index calculus.
Can cryptanalyze larger $n$.

t
ormance ratio
of
$\overline{n}$),
$c$.

method,
atic sieve:

nd $\sqrt{n}$.

e: power of
$\overline{\log n)^2}$).
ntegers.

## Discrete logarithms

Can use the same techniques
to compute $k$ given $3^k \bmod n$.
"Index-calculus methods."

Exponential in $\log n$
to compute discrete logarithm
by collisions, kangaroos, etc.

Subexponential in $\log n$
to use index calculus.
Can cryptanalyze larger $n$.

Collisions, kangaro
work for elliptic cu
so we can cryptana
small elliptic curve

We don't know an
Index calculus doe
work for elliptic cu

Diffie-Hellman spe
use elliptic curves.
Signature-verificat
still use RSA/Rabi

## Discrete logarithms

Can use the same techniques to compute $k$ given $3^k \bmod n$. "Index-calculus methods."

Exponential in $\log n$ to compute discrete logarithm by collisions, kangaroos, etc.

Subexponential in $\log n$ to use index calculus. Can cryptanalyze larger $n$.

Collisions, kangaroos, etc. work for elliptic curves, so we can cryptanalyze small elliptic curves.

We don't know anything better. Index calculus doesn't work for elliptic curves.

Diffie-Hellman speed records use elliptic curves. Signature-verification speed records still use RSA/Rabin variants.