Motivating problem:
Given elliptic curve $E$, integer $n$, and point $P$ on $E$, compute $nP$ on $E$
as quickly as possible.

Many variations of problem. Some applications reuse one $n$
for many $P$’s. Some applications don’t.
Some applications use secret $n$;
must not leak $n$ through timing. Some applications use public $n$. Etc.
Motivating problem:
Given elliptic curve $E$, integer $n$, and point $P$ on $E$, compute $nP$ on $E$ as quickly as possible.

Many variations of problem.
Some applications reuse one $n$ for many $P$'s.
Some applications don't.
Some applications use secret $n$; must not leak $n$ through timing.
Some applications use public $n$.
Etc.

1987 Montgomery:
Focus on large-characteristic curves $y^2 = x^3 + 1$ with small $a \in \{0, 1\}$.
Use pair $(x, z)$ to represent point $P = (x/z, \ldots)$.
Computing $Q, R, \ldots$ takes 6 mults.
Only 5 mults if $G$ has small denominator.
Only 4 mults if $G$ has small numerator and small denominator.
Only 4 mults if $G$...
Motivating problem:
Given elliptic curve $E$, integer $n$, and point $P$ on $E$, compute $nP$ on $E$ as quickly as possible.

Many variations of problem.
Some applications reuse one $n$ for many $P$’s.
Some applications don’t.
Some applications use secret $n$; must not leak $n$ through timing.
Some applications use public $n$. Etc.

1987 Montgomery:
Focus on large-characteristic curves $y^2 = x^3 + ax^2 + x$ with small $a \in \{6, 10, 14, \ldots\}$.

Use pair $(x, z)$ to represent point $P = (x/z, \ldots)$.
Computing $Q, R, Q - R \mapsto Q + R$ takes 6 mults.
Only 5 mults if $Q - R$ has small denominator.
Only 4 mults if $Q - R$ has small numerator and small denominator.
Only 4 mults if $Q = R$. 

Motivating problem:
Given elliptic curve $E$, integer $n$, and point $P$ on $E$, compute $nP$ on $E$ as quickly as possible.

Many variations of problem.
Some applications reuse one $n$ for many $P$'s.
Some applications don't.
Some applications use secret $n$; must not leak through timing.
Some applications use public $n$.

1987 Montgomery:

Focus on large-characteristic curves $y^2 = x^3 + ax^2 + x$ with small $a \in \{6, 10, 14, \ldots\}$. Use pair $(x, z)$ to represent point $P = (x/z, \ldots)$. Computing $Q, R, Q - R \mapsto Q + R$ takes 6 mults.

Only 5 mults if $Q - R$ has small denominator.

Only 4 mults if $Q - R$ has small numerator and small denominator.

Only 4 mults if $Q = R$.

Given $n$, write $nP$ as composition of $Q, R, Q - R \mapsto Q + R$.

E.g. $n = 10$: compute $P, P, 0 \mapsto 2P$; $2P, P, P \mapsto 3P$; $3P, 2P, P \mapsto 5P$; $5P, 5P, 0 \mapsto 10P$.

Overall 20 mults.

Only 18 mults if $P$ has small denominator.

Only 16 mults if $P$ has small numerator and small denominator.
1987 Montgomery:

Focus on large-characteristic curves \( y^2 = x^3 + ax^2 + x \)
with small \( a \in \{6, 10, 14, \ldots\} \).

Use pair \((x, z)\) to represent point \( P = (x/z, \ldots) \).

Computing \( Q, R, Q - R \leftrightarrow Q + R \) takes 6 mults.
Only 5 mults if \( Q - R \) has small denominator.
Only 4 mults if \( Q - R \) has small numerator and small denominator.
Only 4 mults if \( Q = R \).

Given \( n \), write \( P \mapsto nP \) as composition of additions
\( Q, R, Q - R \mapsto Q + R \).

e.g. \( n = 10 \): compute
\( P, \ P, 0 \mapsto 2P \) with 4 mults;
\( 2P, \ P, P \mapsto 3P \) with 6 mults;
\( 3P, 2P, P \mapsto 5P \) with 6 mults;
\( 5P, 5P, 0 \mapsto 10P \) with 4 mults.

Overall 20 mults for \( P \mapsto 10P \).
Only 18 mults if \( P \) has small denominator.
Only 16 mults if \( P \) has small numerator and small denominator.
Given \( n \), write \( P \mapsto nP \) as composition of additions \( Q, R, Q - R \mapsto Q + R \).

e.g. \( n = 10 \): compute

\[
P, \ P, \ 0 \mapsto 2P \text{ with 4 mults};
2P, \ P, \ P \mapsto 3P \text{ with 6 mults};
3P, 2P, \ P \mapsto 5P \text{ with 6 mults};
5P, 5P, \ 0 \mapsto 10P \text{ with 4 mults}.
\]

Overall 20 mults for \( P \mapsto 10P \).

Only 18 mults if \( P \) has small denominator.

Only 16 mults if \( P \) has small numerator and small denominator.

0, \( P, 2P, 3P, 5P, 10P \) is a differential addition chain starting from 0, \( P \), each subsequent term \( Q \) is \( Q + R \) for some \( R \), \( Q, R, Q - R \) already computed.

0, 1, 2, 3, 5, 10 is a differential addition chain starting from 0, 1.

Question: Given \( n \), how to find short differential addition chain starting from 0, 1?

Variations: measure shortness by mults, CPU cycles, etc.
Given $n$, write $P \mapsto nP$ as composition of additions
$Q, R, Q - R \mapsto Q + R$.

e.g. $n = 10$: compute
$P, P, 0 \mapsto 2P$ with 4 mults;
$2P, P, P \mapsto 3P$ with 6 mults;
$3P, 2P, P \mapsto 5P$ with 6 mults;
$5P, 5P, 0 \mapsto 10P$ with 4 mults.
Overall 20 mults for $P \mapsto 10P$.
Only 18 mults
if $P$ has small denominator.
Only 16 mults
if $P$ has small numerator and small denominator.

0, $P, 2P, 3P, 5P, 10P$ is a differential addition chain
starting from 0, $P$:
each subsequent term is $Q + R$ for some $Q, R, Q - R$ already in chain.

0, 1, 2, 3, 5, 10 is a differential addition chain
starting from 0, 1.

Question: Given $n$, how to find short differential addition chain
starting from 0, 1 and ending $n$?

Variations: measure shortness by mults, CPU cycles, etc.
Given $nP$, write as composition of additions $Q + R$.

Compute $2P$ with 4 mults; $3P$ with 6 mults; $5P$ with 6 mults; $10P$ with 4 mults.

Overall 20 mults for $10P$.

Only 18 mults if $P$ has small denominator.

Only 16 mults if $P$ has small numerator and small denominator.

$0, P, 2P, 3P, 5P, 10P$ is a differential addition chain starting from $0, P$: each subsequent term is $Q + R$ for some $Q, R, Q - R$ already in chain.

$0, 1, 2, 3, 5, 10$ is a differential addition chain starting from $0, 1$.

Question: Given $n$, how to find short differential addition chain starting from $0, 1$ and ending $n$?

Variations: measure shortness by mults, CPU cycles, etc.

The binary method:

Obtain $n, n + 1$ from $[n/2], [n/2] + 1$ using one addition with difference 1, one addition with difference 0.

E.g.

$13P, 13P, 0 \mapsto 26P$ with 4 mults;

$14P, 13P, P \mapsto 27P$ with 5 mults,

if $P$ has small denominator.

Overall 9 mults for each bit of $n$, if $P$ has small denominator.
$0, P, 2P, 3P, 5P, 10P$ is a differential addition chain starting from $0, P$: each subsequent term is $Q + R$ for some $Q, R, Q - R$ already in chain.

$0, 1, 2, 3, 5, 10$ is a differential addition chain starting from $0, 1$.

Question: Given $n$, how to find short differential addition chain starting from $0, 1$ and ending $n$?

Variations: measure shortness by mults, CPU cycles, etc.

The binary method: obtain $n, n + 1$ from $\lfloor n/2 \rfloor, \lceil n/2 \rceil + 1$ using one addition with difference 1, one addition with difference 0.

E.g.

$13P, 13P, 0 \leftrightarrow 26P$ with 4 mults;
$14P, 13P, P \leftrightarrow 27P$ with 5 mults, if $P$ has small denominator.

Overall 9 mults for each bit of $n$, if $P$ has small denominator.
$10P$ is a differential addition chain starting from 0:
each subsequent term is $+$ for some term already in chain.

The binary method:
obtain $n, n + 1$ from $\lfloor n/2 \rfloor, \lfloor n/2 \rfloor + 1$ using
one addition with difference 1, one addition with difference 0.
e.g.
$13P, 13P, 0 \rightarrow 26P$ with 4 mults;
$14P, 13P, P \rightarrow 27P$ with 5 mults,
if $P$ has small denominator.
Overall 9 mults for each bit of $n$,
if $P$ has small denominator.

Experiments for average 128-bit find length $\approx 1.5$ per bit,
instead of 2 per bit.
Lower bound $\approx 1$.
Count mults instead of length:
$\approx 8.885$ per bit,
instead of 9 per bit.
Disadvantages: harder to find;
no uniform structure; harder to avoid leaking $n$ through timing.
The binary method:

obtain \( n, n + 1 \) from

\[ \lfloor n/2 \rfloor, \lfloor n/2 \rfloor + 1 \]

using

one addition with difference 1,
one addition with difference 0.

For example,

\( 13P, 13P, 0 \rightarrow 26P \) with 4 mults;
\( 14P, 13P, P \rightarrow 27P \) with 5 mults,

if \( P \) has small denominator.

Overall 9 mults for each bit of \( n \),

if \( P \) has small denominator.

1992 Montgomery,
1996 Bleichenbacher,
2001 Tsuruoka: Can do better!

Experiments for average 128-bit \( n \):

find length \( \approx 1.533 \) per bit,

instead of 2 per bit.

Lower bound \( \approx 1.440 \) per bit.

Count mults instead of length:

\( \approx 8.885 \) per bit,

instead of 9 per bit.

Disadvantages: harder to find;
no uniform structure; harder to avoid leaking \( n \) through timing.
The binary method:

obtain + 1 from 2

2 + 1 using one addition with difference 1,
one addition with difference 0.

e.g.

13

13

0

0

26

with 4 mults;

14

13

0

0

27

with 5 mults,

if

has small denominator.

Overall 9 mults

for each bit of

, if

has small denominator.

1992 Montgomery,

1996 Bleichenbacher,

2001 Tsuruoka: Can do better!

Experiments for average 128-bit

find length \( \approx 1.533 \) per bit,

instead of 2 per bit.

Lower bound \( \approx 1.440 \) per bit.

Count mults instead of length:

\( \approx 8.885 \) per bit,

instead of 9 per bit.

Disadvantages: harder to find;

no uniform structure; harder to
avoid leaking \( n \) through timing.

Two-dimensional question:

Given \( m, n \), how to find short differential addition chain

starting from the vectors

(0, 0), (1, 0), (0, 1), and ending \((m, n)\)?

Motivating problem:

Given elliptic curve

, integers \( m, n \),

and points \( P, Q \),
compute \( mP + nQ \) as quickly as possible.
1992 Montgomery,
1996 Bleichenbacher,
2001 Tsuruoka: Can do better!

Experiments for average 128-bit \( n \)
find length \( \approx 1.533 \) per bit,
instead of 2 per bit.
Lower bound \( \approx 1.440 \) per bit.

Count multis instead of length:
\( \approx 8.885 \) per bit,
instead of 9 per bit.

Disadvantages: harder to find;
no uniform structure; harder to
avoid leaking \( n \) through timing.

Two-dimensional question:
Given \( m, n \), how to find
short differential addition chain
starting from the vectors
\((0, 0), (1, 0), (0, 1), (1, -1)\)
and ending \((m, n)\)?

Motivating problem:
Given elliptic curve \( E \),
integers \( m, n \),
and points \( P, Q, P - Q \),
compute \( mP + nQ \) on \( E \)
as quickly as possible.
1992 Montgomery,
1996 Bleichenbacher,
2001 Tsuruoka: Can do better!
Experiments for average 128-bit
find length
1.533 per bit, instead of 2 per bit.
Lower bound
440 per bit.
Count mults instead of length:
885 per bit, instead of 9 per bit.
Disadvantages: harder to find;
no uniform structure; harder to
avoid leaking through timing.

Two-dimensional question:
Given \( m, n \), how to find
short differential addition chain
starting from the vectors
\((0, 0), (1, 0), (0, 1), (1, -1)\)
and ending \((m, n)\)?

Motivating problem:
Given elliptic curve \( E \),
integers \( m, n \),
and points \( P, Q, P - Q \),
compute \( mP + nQ \) on \( E \)
as quickly as possible.

<table>
<thead>
<tr>
<th>dim</th>
<th>method</th>
<th>mults</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>easy binary</td>
<td>19.000</td>
</tr>
<tr>
<td>2</td>
<td>Schoenmakers</td>
<td>17.250</td>
</tr>
<tr>
<td>2</td>
<td>Akishita</td>
<td>14.250</td>
</tr>
<tr>
<td>2</td>
<td>new binary</td>
<td>14.000</td>
</tr>
<tr>
<td>2</td>
<td>Montgomery</td>
<td>10.261</td>
</tr>
<tr>
<td>2</td>
<td>new ext gcd</td>
<td>9.918</td>
</tr>
<tr>
<td>1</td>
<td>easy binary</td>
<td>9.000</td>
</tr>
<tr>
<td>1</td>
<td>standard</td>
<td>8.885</td>
</tr>
<tr>
<td></td>
<td>Fibonacci case</td>
<td>8.643</td>
</tr>
</tbody>
</table>
Two-dimensional question: Given $m, n$, how to find short differential addition chain starting from the vectors $(0, 0), (1, 0), (0, 1), (1, -1)$ and ending $(m, n)$?

Motivating problem: Given elliptic curve $E$, integers $m, n$, and points $P, Q, P - Q$, compute $mP + nQ$ on $E$ as quickly as possible.

For average 128-bit exponents, small $P, Q, P - Q$ denominators:

<table>
<thead>
<tr>
<th>dim</th>
<th>method</th>
<th>mults per bit</th>
<th>unif</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>easy binary</td>
<td>19.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Schoenmakers</td>
<td>17.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>Akishita</td>
<td>14.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new binary</td>
<td>14.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Montgomery</td>
<td>10.261</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new ext gcd</td>
<td>9.918</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>easy binary</td>
<td>9.000</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>standard</td>
<td>8.885</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>Fibonacci case</td>
<td>8.643</td>
<td></td>
</tr>
</tbody>
</table>
Two-dimensional question: Given \((0,0), (1,0), (0,1), (1,1)\), how to find short differential addition chain starting from the vectors \((0,0), (1,0), (0,1), (1,1)\)?

Motivating problem: Given elliptic curve \(E\), integers \(n\) and points \(P, Q, P - Q\), compute \(nPQ\) on \(E\) as quickly as possible.

For average 128-bit exponents, small \(P, Q, P - Q\) denominators:

<table>
<thead>
<tr>
<th>dim</th>
<th>method</th>
<th>mults per bit</th>
<th>unif</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>easy binary</td>
<td>19.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Schoenmakers</td>
<td>17.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>Akishita</td>
<td>14.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new binary</td>
<td>14.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Montgomery</td>
<td>10.261</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new ext gcd</td>
<td>9.918</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>easy binary</td>
<td>9.000</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>standard</td>
<td>8.885</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>Fibonacci case</td>
<td>8.643</td>
<td></td>
</tr>
</tbody>
</table>

Easy dim-2 binary chain:
(0, 0)  (0, 1)
↓ ↓
(1, 0)  (1, 1)
↓ ↓
(2, 1)  (2, 2)
↓ ↓
(4, 3)  (4, 4)
↓ ↓
(9, 7)  (9, 8)
↓ ↓
(18, 14) (18, 15)
For average 128-bit exponents, small $P, Q, P - Q$ denominators:

<table>
<thead>
<tr>
<th>dim</th>
<th>method</th>
<th>mults per bit</th>
<th>unif</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>easy binary</td>
<td>19.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Schoenmakers</td>
<td>17.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>Akishita</td>
<td>14.250</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new binary</td>
<td>14.000</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Montgomery</td>
<td>10.261</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>new ext gcd</td>
<td>9.918</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>easy binary</td>
<td>9.000</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>standard</td>
<td>8.885</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>Fibonacci case</td>
<td>8.643</td>
<td></td>
</tr>
</tbody>
</table>

Easy dim-2 binary chain:

```
(0, 0) (0, 1) (1, 0) (1, -1)
\downarrow \downarrow \downarrow \downarrow
(1, 0) (1, 1) (2, 0) (2, 1)
\downarrow \downarrow \downarrow \downarrow
(2, 1) (2, 2) (3, 1) (3, 2)
\downarrow \downarrow \downarrow \downarrow
(4, 3) (4, 4) (5, 3) (5, 4)
\downarrow \downarrow \downarrow \downarrow
(9, 7) (9, 8) (10, 7) (10, 8)
\downarrow \downarrow \downarrow \downarrow
(18, 14) (18, 15) (19, 14) (19, 15)
```
For average 128-bit exponents, small denominators:

<table>
<thead>
<tr>
<th>Dim Method</th>
<th>Mults Per Bit</th>
<th>Unif</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy Binary</td>
<td>19.000</td>
<td>yes</td>
</tr>
<tr>
<td>Montgomery</td>
<td>17.250</td>
<td>no</td>
</tr>
<tr>
<td>Akishita</td>
<td>14.250</td>
<td>no</td>
</tr>
<tr>
<td>New Binary</td>
<td>14.000</td>
<td>yes</td>
</tr>
<tr>
<td>Standard</td>
<td>10.261</td>
<td>no</td>
</tr>
<tr>
<td>New Ext Gcd</td>
<td>9.918</td>
<td>no</td>
</tr>
<tr>
<td>Easy Binary</td>
<td>9.000</td>
<td>yes</td>
</tr>
<tr>
<td>Standard</td>
<td>8.885</td>
<td>no</td>
</tr>
<tr>
<td>Fibonacci Case</td>
<td>8.643</td>
<td></td>
</tr>
</tbody>
</table>

Easy dim-2 binary chain:

(0,0)  (0,1)  (1,0)  (1,-1)
↓     ↓     ↓     ↓
(1,0)  (1,1)  (2,0)  (2,1)
↓     ↓     ↓     ↓
(2,1)  (2,2)  (3,1)  (3,2)
↓     ↓     ↓     ↓
(4,3)  (4,4)  (5,3)  (5,4)
↓     ↓     ↓     ↓
(9,7)  (9,8)  (10,7) (10,8)
↓     ↓     ↓     ↓
(18,14) (18,15) (19,14) (19,15)

New dim-2 binary chain:

(0,0)  (1,0)
↓     ↓
(1,1)  (2,0)
↓     ↓
(3,1)  (2,2)
↓     ↓
(5,3)  (4,4)
↓     ↓
(9,7)  (10,8)
↓     ↓
(19,15) (18,14)
Easy dim-2 binary chain:

(0, 0) (0, 1) (1, 0) (1, -1)
(1, 0) (1, 1) (2, 0) (2, 1)
(2, 1) (2, 2) (3, 1) (3, 2)
(4, 3) (4, 4) (5, 3) (5, 4)
(9, 7) (9, 8) (10, 7) (10, 8)
(18, 14) (18, 15) (19, 14) (19, 15)

New dim-2 binary chain:

(0, 0) (1, 0) (0, 1) (1, -1)
(1, 1) (2, 0) (2, 1)
(3, 1) (2, 2) (3, 2)
(5, 3) (4, 4) (5, 4)
(9, 7) (10, 8) (9, 8)
(19, 15) (18, 14) (18, 15)
Line in easy binary chain has \((a, b), (a, b + 1), (a + 1, b + 1)\). Obtain next line by double-add-advance.

New observation. A line can be even, odd or (odd, even) or (even, even) chosen recursively. If next line can be obtained by double-add-advance, only 14 mults if \(P, Q, R\) have small denominators.

New dim-2 binary chain:

\[(0, 0) (1, 0) (0, 1) (1, -1)\]

\[(1, 1) (2, 0) (2, 1)\]

\[(3, 1) (2, 2) (3, 2)\]

\[(5, 3) (4, 4) (5, 4)\]

\[(9, 7) (10, 8) (9, 8)\]

\[(19, 15) (18, 14) (18, 15)\]

Line in easy binary chain has \((a, b), (a, b + 1), (a + 1, b), (a + 1, b + 1)\). Obtain next line by double-add-add-add.

New observation: can omit (even, odd) or (odd, even), chosen recursively so that next line can be obtained by double-add-add-add.

14 mults if \(P, Q, P - Q\) have small denominators.

Intermediate results: 2000

Schoenmakers, 2001 Akishita.
How to do better than binary?

Don't worry about uniformity.

Critical idea for dim 1:
Build chain 0, 1, 2, 3, 5, ... by choosing $r \approx \frac{\sqrt{5} - 1}{2}$ and building chain 0, 1, ..., $r$, $n - r$, $r$.

Try many $r$'s, keep best.

Some further choices here:
could build \{r, n - 2r, 2r\} and building chain \{r, n/2 - r, r\}.

14 mults if $P, Q, P - Q$ have small denominators.

Intermediate results: 2000
Schoenmakers, 2001 Akishita.
Line in easy binary chain has \((a, b), (a, b + 1), (a + 1, b), (a + 1, b + 1)\). Obtain next line by double-add-add-add.

New observation: can omit (even, odd) or (odd, even), chosen recursively so that next line can be obtained by double-add-add-add.

14 mults if \(P, Q, P - Q\) have small denominators.


How to do better than binary? Don't worry about uniformity.

Critical idea for dim 1:
Build chain \(0, 1, \ldots, n\) by choosing \(r \approx n(\sqrt{5} - 1)/2\) and building chain \(0, 1, \ldots, r, n - r, n\). Try many \(r\)'s, keep best.

Some further choices here: could build \(\{r, n - r, n\}\) from \(\{r, n - 2r, n - r\}\) or from \(\{n - r, 2r - n, r\}\) or from \(\{r, n/2 - r, n/2\}\) or \ldots.
How to do better than binary? Don’t worry about uniformity.

Critical idea for dim 1:
Build chain $0, 1, \ldots, n$
by choosing $r \approx n(\sqrt{5} - 1)/2$
and building chain
$0, 1, \ldots, r, n - r, n$.

Try many $r$'s, keep best.

Some further choices here:
could build $\{r, n - r, n\}$
from $\{r, n - 2r, n - r\}$ or
from $\{n - r, 2r - n, r\}$ or
from $\{r, n/2 - r, n/2\}$ or ....

e.g. $n = 100$, $r = 39$: Build chain
$0, 1, 2, 3, 5, 7, 12, \ldots$ by building $\{39, 61\}$
from $\{22, 39, 61\}$

What about dim 2?
Obvious adaptation of idea:
Build chain $\ldots, (q, r), (m - r, m - q)$
by choosing $(q, r)$
and building chain $\ldots, (q, r), (m - r, m - q)$

Results: 2000 2001 Akishita.
How to do better than binary?
Don't worry about uniformity.

Critical idea for dim 1:
Build chain $0, 1, \ldots, n$ by choosing $r \approx n(\sqrt{5} - 1)/2$ and building chain $0, 1, \ldots, r, n - r, n$.
Try many $r$'s, keep best.

Some further choices here:
could build $\{r, n - r, n\}$ from $\{r, n - 2r, n - r\}$ or $\{n - r, 2r - n, r\}$ or $\{r, n/2 - r, n/2\}$ or \ldots.

\[\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 5 & 7 & 12 & 17 & 22 & 39 & 61 & 100
\end{array}\]

by building $\{39, 61, 100\}$ from $\{22, 39, 61\}$ etc.

What about dim 2?
Obvious adaptation of idea:
Build chain \ldots, $(m, n)$ by choosing $(q, r)$ and building chain \ldots, $(q, r), (m - q, n - r), (m, n)$.

e.g. $n = 100, r = 39$:
Build chain $0, 1, 2, 3, 5, 7, 12, 17, 22, 39, 61, 100$ by building $\{39, 61, 100\}$ from $\{22, 39, 61\}$ etc.
How to do better than binary?

Don't worry about uniformity.

Critical idea for dim 1:

Build chain 0 1

by choosing

and building chain 0 1.

Try many 's, keep best.

Some further choices here:
could build

from

or

from

or

from

.  

e.g. \( n = 100, \ r = 39 \):

Build chain

0, 1, 2, 3, 5, 7, 12, 17, 22, 39, 61, 100

by building \{39, 61, 100\} from \{22, 39, 61\} etc.

What about dim 2?

Obvious adaptation of idea:

Build chain . . ., \((m, n)\)

by choosing \((q, r)\)

and building chain . . ., \((q, r), (m - q, n - r), (m, n)\).

Hmmm, what's the endgame?

How to build short chain with \((8, 16)\)?

Several plausible approaches, but all of them scale badly.

Normally this construction is abandoned.

e.g. Work backwards from \((314, 271)\) and \((194, 167)\) to \((120, 104)\), then \((74, 63)\), then \((46, 41)\), then \((28, 22)\), then \((18, 19)\), then \((10, 3)\).

...
e.g. \( n = 100, r = 39 \): Build chain
0, 1, 2, 3, 5, 7, 12, 17, 22, 39, 61, 100
by building \{39, 61, 100\}
from \{22, 39, 61\} etc.

What about dim 2?
Obvious adaptation of idea:
Build chain \ldots, (m, n)\nby choosing (q, r)
and building chain
\ldots, (q, r), (m - q, n - r), (m, n).

e.g. Work backwards from
(314, 271) and (194, 167) to
(120, 104), then (74, 63), then
(46, 41), then (28, 22), then
(18, 19), then (10, 3), then
(8, 16).

Hmmm, what’s the endgame?
How to build short chain with
\{(8, 16), (10, 3), (18, 19)\}?
Several plausible approaches,
but all of them scale badly.
Normally this construction
is abandoned.
e.g. Work backwards from
(314, 271) and (194, 167) to
(120, 104), then (74, 63), then
(46, 41), then (28, 22), then
(18, 19), then (10, 3), then
(8, 16).

Hmmm, what’s the endgame?
How to build short chain with
\{(8, 16), (10, 3), (18, 19)\}?

Several plausible approaches,
but all of them scale badly.

Normally this construction
is abandoned.

New observation:
Simple endgames work well
if \( rm - qn = \Delta \) with, e.g., \( \Delta = 2^3 \).

Often find very good chains.

Easy to find \((q, r)\) given \((m, n, \Delta)\):
standard ext-gcd computation.

What if \((m, n)\) not coprime?
Great! Exploit factor.

Try many good choices
for \((\Delta, q, r)\), keep best.

\( m = 39: \)
\{17, 22, 39, 61, 100 \}
- etc.

\( \Delta = \Delta \) etc.

\( \Delta \) etc.
e.g. Work backwards from (314, 271) and (194, 167) to (120, 104), then (74, 63), then (46, 41), then (28, 22), then (18, 19), then (10, 3), then (8, 16).

Hmmm, what’s the endgame?
How to build short chain with \{ (8, 16), (10, 3), (18, 19) \}?

Several plausible approaches, but all of them scale badly.
Normally this construction is abandoned.

New observation:
Simple endgames work well if \( rm -qn = \Delta \)
with, e.g., \( \Delta = \pm 2^a 3^b \).
Often find very good chains.

Easy to find \((q, r)\) given \((m, n, \Delta)\):
standard ext-gcd computation.

What if \((m, n)\) not coprime?
Great! Exploit factor.
Try many good choices for \((\Delta, q, r)\), keep best.
New observation:
Simple endgames work well if $rm - qn = \Delta$
with, e.g., $\Delta = \pm 2^a 3^b$.
Often find very good chains.

Easy to find $(q, r)$
given $(m, n, \Delta)$:
standard ext-gcd computation.

What if $(m, n)$ not coprime?
Great! Exploit factor.

Try many good choices
for $(\Delta, q, r)$, keep best.

Example of new chain:
$(0, 0), (1, 0), (0, 1), (1, 1), (1, 2), (2, 3), (3, 5), (4, 7), (5, 9), (9, 16), (14, 25), (19, 34), (33, 59), (66, 118), (71, 127), (132, 236), (203, 379), (325, 581), (528, 985), (731, 1307), (1259, 2251), (1787, 3195), (2518, 4502), (3249, 5809), (5036, 9004), (6823, 12199), (10072, 18008), (16895, 30207), (26967, 48215),...
New observation:
Simple endgames work well
if \( rm - qn = \Delta \)
with, e.g., \( \Delta = \pm 2^a 3^b \).
Often find very good chains.

Easy to find \((q, r)\)
given \((m, n, \Delta)\):
standard ext-gcd computation.

What if \((m, n)\) not coprime?
Great! Exploit factor.

Try many good choices
for \((\Delta, q, r)\), keep best.

Example of new chain:
\((0, 0), (1, 0), (0, 1), (1, -1), (1, 1), (1, 2), (2, 3), (3, 5), (4, 7), (5, 9), (9, 16), (14, 25), (19, 34), (33, 59), (38, 68), (66, 118), (71, 127), (61, 109), (132, 236), (203, 363), (264, 472), (325, 581), (528, 944), (731, 1307), (1259, 2251), (1787, 3195), (2518, 4502), (3249, 5809), (5036, 9004), (6823, 12199), (10072, 18008), (16895, 30207), (26967, 48215).\)