Proiect SDA

Dragos-Gabriel Danciulescu

May 2022

Example Exercise

Use the baby-step giant-step method to find x such that

$$5^x \equiv 107 \bmod 179$$

. Pick m = $\lfloor\sqrt{89}\rfloor$.Copy, extend and fill the appropriate tables.

a. We will create the Baby steps table in which $g = 5$ for i up to 9:

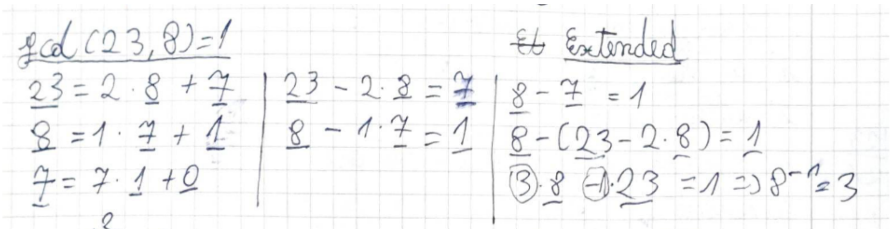
i	0	1	2	3	4	5	6	7	8	9
5^i mod 179	1	5	25	125	88	82	52	81	47	56

Next, we note that:
$g^{-\sqrt{89}} = 5^{-9} = 56^{-1} = 16$ mod 179
So, $h \cdot g^{-mj} = 107 \cdot 16^j$
Now,we will create the Giant steps table. Note that in the algorithm, these values are not stored in memory like the ones in the first table.

j	0	1	2
$107 \cdot 16^j$ mod 179	107	101	5

We can now observe that we've found a match, for $i = 1$ and $j = 2$.
So,following the theory:
$g^{i+m\cdot j} = h$
$5^{1+9\cdot 2} = 107$
5^{19} mod $179 = 107$
We've also checked the value with Wolfram Alpha, it is correct, so:
$x = 19$

Notebook Examples and Theory



$\underline{\gcd(23, 8) = 1}$

$23 = 2\cdot 8 + 7$

$8 = 1\cdot 7 + 1$

$7 = 7\cdot 1 + 0$

$23 - 2\cdot 8 = 7$

$8 - 1\cdot 7 = 1$

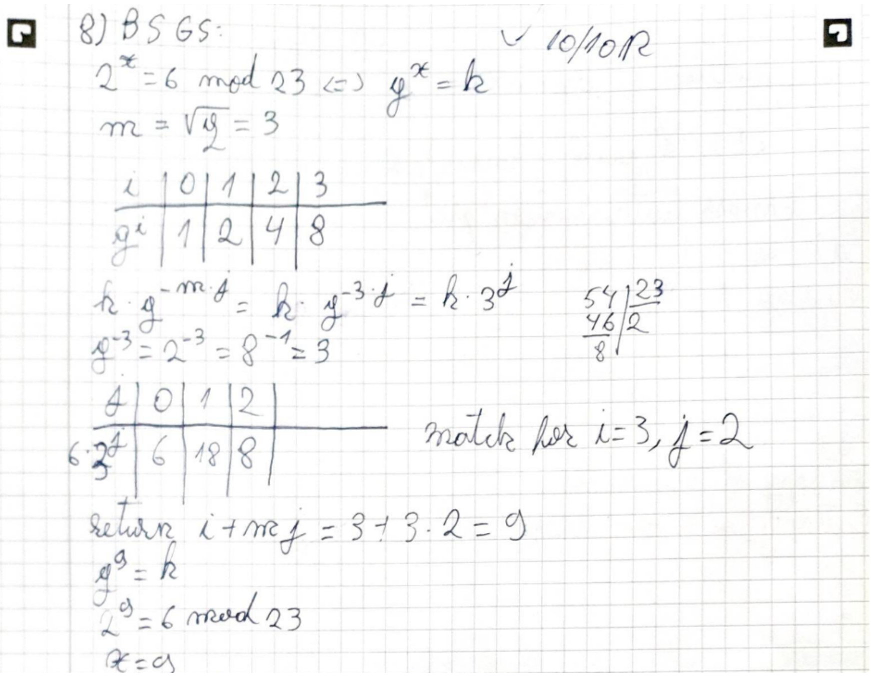
~~Et~~ Extended

$8 - 7 = 1$

$8 - (23 - 2\cdot 8) = 1$

$\textcircled{3}\cdot 8 \; \textcircled{-1}\cdot 23 = 1 \Rightarrow 8^{-1} = 3$

Figure 1: Extended Euclidean algorithm notebook example



8) BSGS:

✓ 10/10R

$2^x = 6 \mod 23 \Leftrightarrow g^x = k$

$m = \sqrt{\underset{2}{g}} = 3$

i	0	1	2	3
g^i	1	2	4	8

$k\cdot g^{-m\cdot j} = k\cdot g^{-3\cdot j} = k\cdot 3^j$

54 | 23
46 | 2
8

$g^{-3} = 2^{-3} = 8^{-1} = 3$

j	0	1	2
$6\cdot 3^j$	6	18	8

match for $i = 3, j = 2$

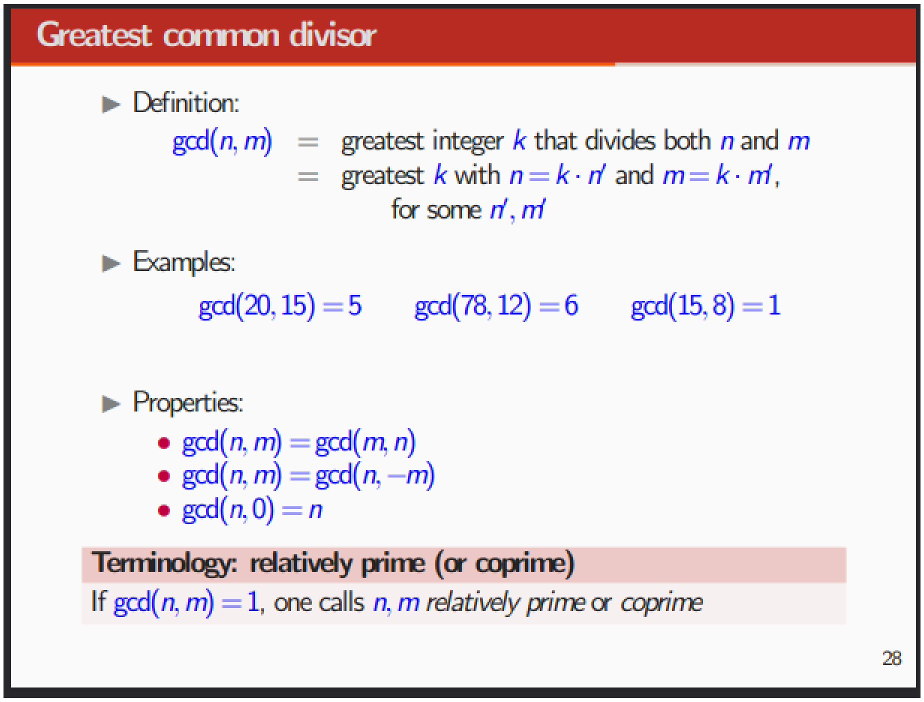
return $i + mj = 3 + 3\cdot 2 = 9$

$g^9 = k$

$2^9 = 6 \mod 23$

$x = 9$

Figure 2: Baby-Step Giant-Step notebook example



Greatest common divisor

- Definition:

$$\begin{aligned}\gcd(n,m) &= \text{greatest integer } k \text{ that divides both } n \text{ and } m\\ &= \text{greatest } k \text{ with } n = k \cdot n' \text{ and } m = k \cdot m',\\ &\qquad \text{for some } n', m'\end{aligned}$$

- Examples:

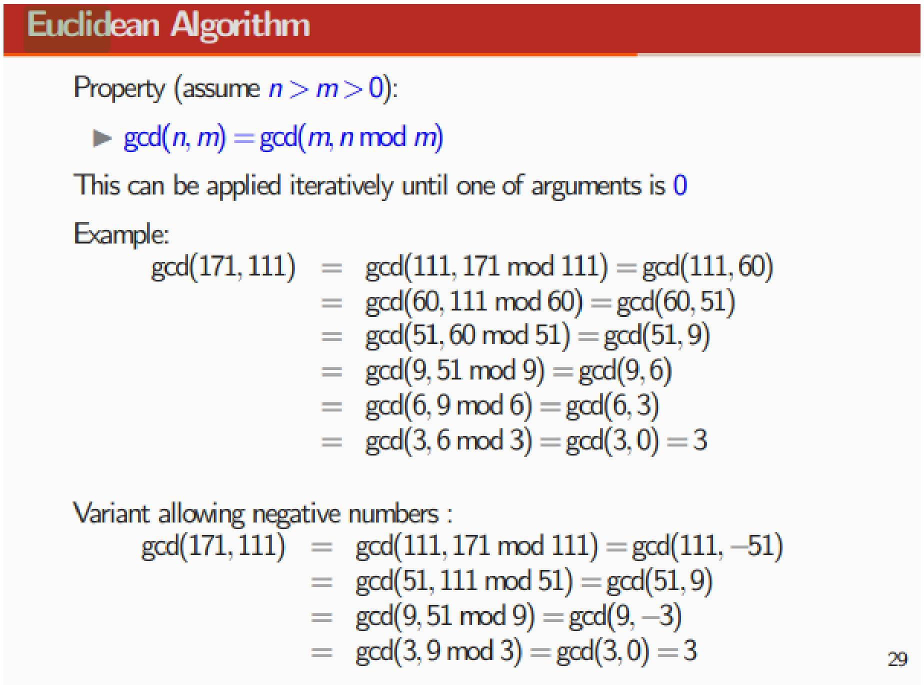
$\gcd(20,15) = 5 \qquad \gcd(78,12) = 6 \qquad \gcd(15,8) = 1$

- Properties:
  - $\gcd(n,m) = \gcd(m,n)$
  - $\gcd(n,m) = \gcd(n,-m)$
  - $\gcd(n,0) = n$

Terminology: relatively prime (or coprime)

If $\gcd(n,m) = 1$, one calls n, m *relatively prime* or *coprime*

Figure 3: Greatest Common divisor



Euclidean Algorithm

Property (assume $n > m > 0$):

- $\gcd(n, m) = \gcd(m, n \bmod m)$

This can be applied iteratively until one of arguments is 0

Example:

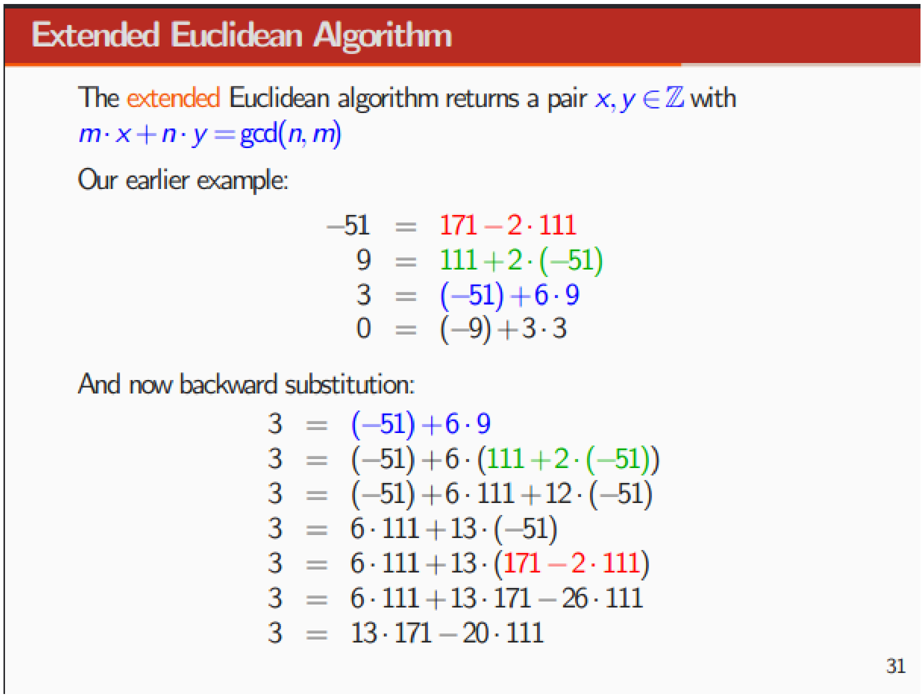
$$\begin{aligned}
\gcd(171, 111) &= \gcd(111, 171 \bmod 111) = \gcd(111, 60)\\
&= \gcd(60, 111 \bmod 60) = \gcd(60, 51)\\
&= \gcd(51, 60 \bmod 51) = \gcd(51, 9)\\
&= \gcd(9, 51 \bmod 9) = \gcd(9, 6)\\
&= \gcd(6, 9 \bmod 6) = \gcd(6, 3)\\
&= \gcd(3, 6 \bmod 3) = \gcd(3, 0) = 3
\end{aligned}$$

Variant allowing negative numbers :

$$\begin{aligned}
\gcd(171, 111) &= \gcd(111, 171 \bmod 111) = \gcd(111, -51)\\
&= \gcd(51, 111 \bmod 51) = \gcd(51, 9)\\
&= \gcd(9, 51 \bmod 9) = \gcd(9, -3)\\
&= \gcd(3, 9 \bmod 3) = \gcd(3, 0) = 3
\end{aligned}$$

29

Figure 4: Euclidean algorithm



Extended Euclidean Algorithm

The extended Euclidean algorithm returns a pair $x, y \in \mathbb{Z}$ with $m \cdot x + n \cdot y = \gcd(n, m)$

Our earlier example:

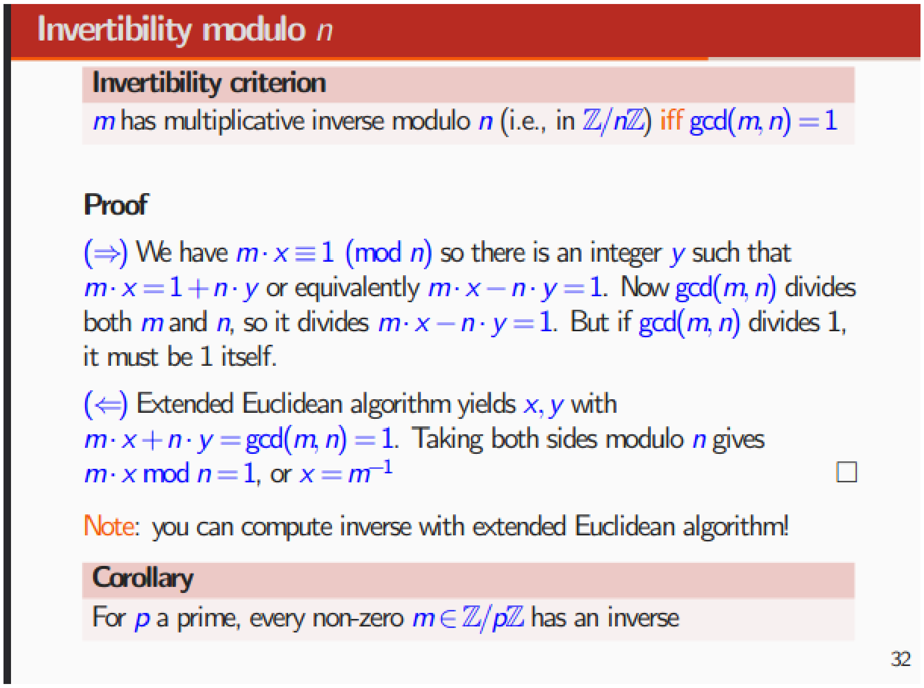
$$\begin{aligned} -51 &= 171 - 2 \cdot 111 \\ 9 &= 111 + 2 \cdot (-51) \\ 3 &= (-51) + 6 \cdot 9 \\ 0 &= (-9) + 3 \cdot 3 \end{aligned}$$

And now backward substitution:

$$\begin{aligned} 3 &= (-51) + 6 \cdot 9 \\ 3 &= (-51) + 6 \cdot (111 + 2 \cdot (-51)) \\ 3 &= (-51) + 6 \cdot 111 + 12 \cdot (-51) \\ 3 &= 6 \cdot 111 + 13 \cdot (-51) \\ 3 &= 6 \cdot 111 + 13 \cdot (171 - 2 \cdot 111) \\ 3 &= 6 \cdot 111 + 13 \cdot 171 - 26 \cdot 111 \\ 3 &= 13 \cdot 171 - 20 \cdot 111 \end{aligned}$$

31

Figure 5: Extended euclidean algorithm



Invertibility modulo n

Invertibility criterion

m has multiplicative inverse modulo n (i.e., in $\mathbb{Z}/n\mathbb{Z}$) iff $\gcd(m,n)=1$

Proof

($\Rightarrow$) We have $m \cdot x \equiv 1 \pmod{n}$ so there is an integer y such that $m \cdot x = 1 + n \cdot y$ or equivalently $m \cdot x - n \cdot y = 1$. Now $\gcd(m,n)$ divides both m and n, so it divides $m \cdot x - n \cdot y = 1$. But if $\gcd(m,n)$ divides 1, it must be 1 itself.

($\Leftarrow$) Extended Euclidean algorithm yields x, y with $m \cdot x + n \cdot y = \gcd(m,n) = 1$. Taking both sides modulo n gives $m \cdot x \bmod n = 1$, or $x = m^{-1}$ □

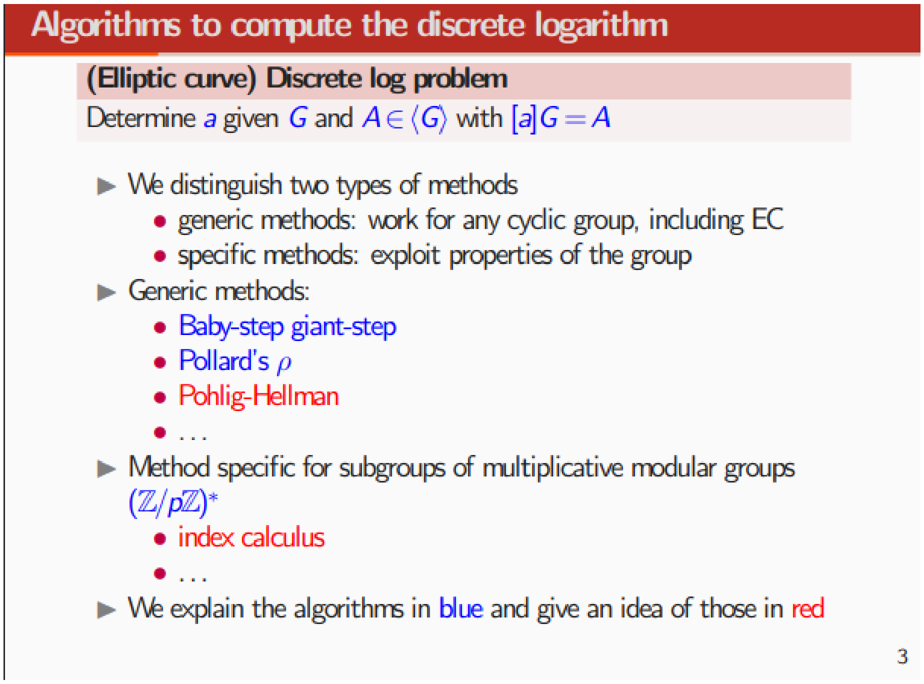
Note: you can compute inverse with extended Euclidean algorithm!

Corollary

For p a prime, every non-zero $m \in \mathbb{Z}/p\mathbb{Z}$ has an inverse

32

Figure 6: Computing inverse using Euclid's algorithm



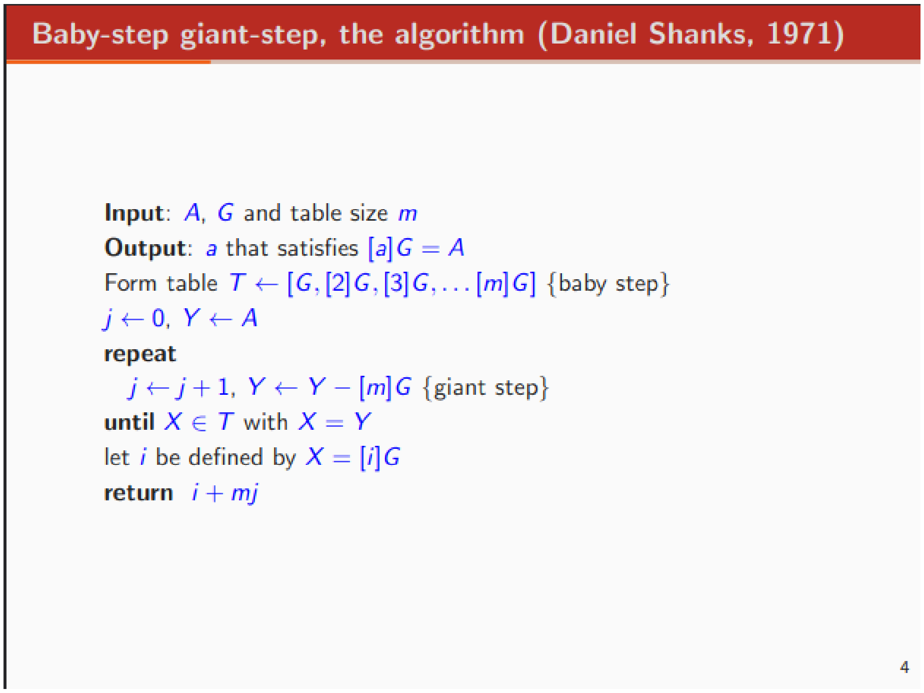
Algorithms to compute the discrete logarithm

(Elliptic curve) Discrete log problem

Determine a given G and $A \in \langle G \rangle$ with $[a]G = A$

- We distinguish two types of methods
  - generic methods: work for any cyclic group, including EC
  - specific methods: exploit properties of the group
- Generic methods:
  - Baby-step giant-step
  - Pollard's ρ
  - Pohlig-Hellman
  - ...
- Method specific for subgroups of multiplicative modular groups $(\mathbb{Z}/p\mathbb{Z})^*$
  - index calculus
  - ...
- We explain the algorithms in blue and give an idea of those in red

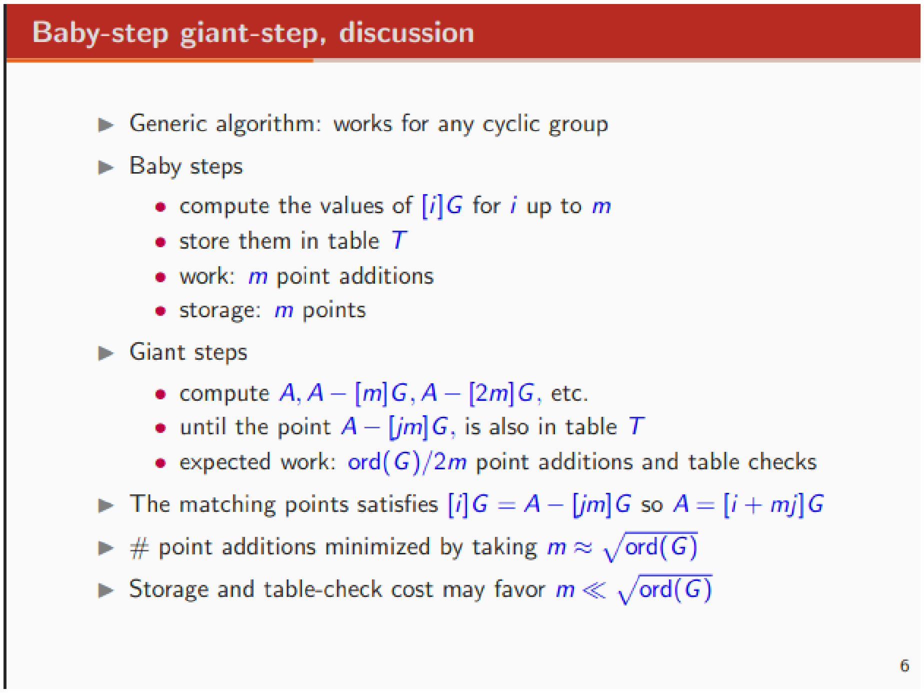
Figure 7: The discrete log problem in cryptography



Baby-step giant-step, the algorithm (Daniel Shanks, 1971)

```
**Input**: $A$, $G$ and table size $m$
**Output**: $a$ that satisfies $[a]G = A$
Form table $T \leftarrow [G, [2]G, [3]G, \dots [m]G]$ {baby step}
$j \leftarrow 0$, $Y \leftarrow A$
**repeat**
  $j \leftarrow j + 1$, $Y \leftarrow Y - [m]G$ {giant step}
**until** $X \in T$ with $X = Y$
let $i$ be defined by $X = [i]G$
**return** $i + mj$
```

Figure 8: Explanation of Baby-Step Giant-Step algorithm



Baby-step giant-step, discussion

- Generic algorithm: works for any cyclic group
- Baby steps
  - compute the values of $[i]G$ for i up to m
  - store them in table T
  - work: m point additions
  - storage: m points
- Giant steps
  - compute $A, A - [m]G, A - [2m]G$, etc.
  - until the point $A - [jm]G$, is also in table T
  - expected work: $\mathrm{ord}(G)/2m$ point additions and table checks
- The matching points satisfies $[i]G = A - [jm]G$ so $A = [i + mj]G$
- # point additions minimized by taking $m \approx \sqrt{\mathrm{ord}(G)}$
- Storage and table-check cost may favor $m \ll \sqrt{\mathrm{ord}(G)}$

Figure 9: A discussion about BSGS