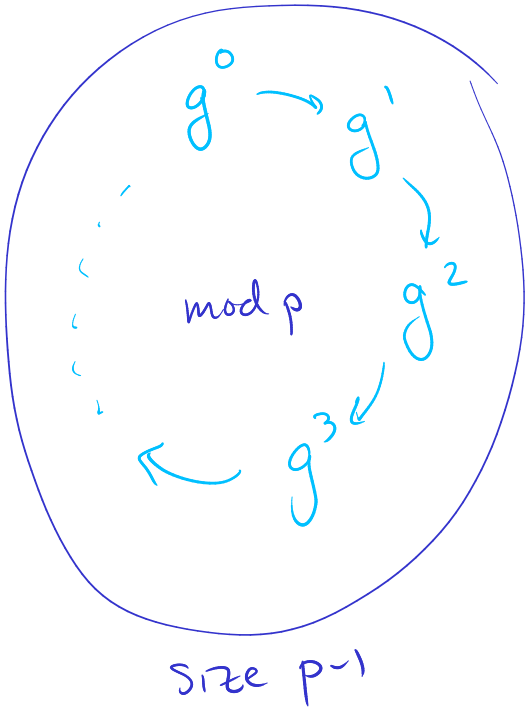


Methods to attack discrete log

$$g^x = h$$

Solve for x given g, h

① Exhaustive search

$g^1, g^2, \ldots$ look for our target h

② If we notice $\boxed{g^a = hg^b}$ then

$g^a = g^x g^b$ x unknown

$\Rightarrow a = x + b$

$\Rightarrow \boxed{x = a - b}$

Solve for x.

If I had 2 lists: $g^1, g^2, \ldots$

$hg, hg^2, \ldots$

look for a collision

naive idea

$\Rightarrow$ get $g^a = hg^b$ & use $x = a - b$.

Baby-Step–Giant-Step

Challenge (DLP): $g^a \equiv h \pmod p$. Given g, h, p find a.

Babystep list: multiply by g repeatedly

$$g, g^2, g^3, g^4, \ldots \pmod p$$

Giantstep list: multiply by g^{-N} repeatedly

$$h, hg^{-N}, hg^{-2N}, hg^{-3N}, \ldots \pmod p$$

If we get a collision (something in both lists) then

$$g^j \equiv hg^{-Nk} \pmod p \qquad j,k \text{ known}$$

$$\Rightarrow g^{j+Nk} \equiv h$$

i.e. $a = j + Nk \pmod{p-1}$ exponents

Choice of N is

$$N = \lceil \sqrt{p-1} \rceil$$

So $N^2 \geq p-1$

then $a = j + kN$ is like a "base N" representation

if $0 \leq j \leq N$, $0 \leq k \leq N$

then $0 < a < N(N+1) \approx N^2 \approx p-1$.

Example: Solve $2^a \equiv 7 \pmod{11}$

$N = \lceil \sqrt{10} \rceil = 4$

$g^{-4} = 2^{-4} = 2^6 = 5 \cdot 4 = 9$

Baby Steps

$2^1 = 2$

$2^2 = 4$

$\boxed{2^3} = 8 = g^3$

$2^4 = 5$

length $\sim N$

Giant Steps

$\boxed{7 \cdot 9} = 63 = 8 = h \cdot g^{-N \cdot 1}$ collision!

$7 \cdot 9^2$

$7 \cdot 9^3$

$7 \cdot 9^4$

$2^3 = 7 \cdot 9$

$2^3 = 7 \cdot 2^{-4 \cdot 1}$

$7 = 2^3 \cdot 2^4 = 2^7$

$a = 7$

check!

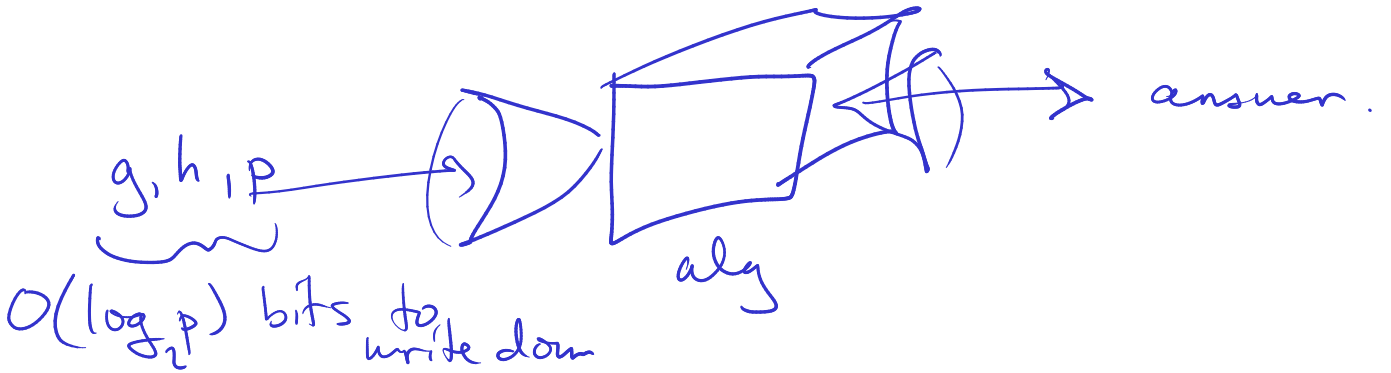
$2^7 = 2^3 2^4 = 8 \cdot 5 = 40 = 7$

Runtime of Exhaustive Search: $O(p)$ steps — $p-1$ steps to list

Baby-Step-Giant-Step: 2 lists of size $\sqrt{p}$ — $O(\sqrt{p})$ steps.

Both exponential time.

Runtimes are measured in terms of input size of the problem



A runtime of $O(p)$ is exponential in $\log p$.

$$= O(2^{\log_2 p})$$

$$O(\sqrt{p}) = O(2^{\frac{1}{2}\log_2 p})$$

is also exponential in $\log_2 p$.

These are exponential algorithms.

Successive squaring required $O(\log_2 p)$ steps

= polynomial in $\log_2 p$

So modular exponentiation is polynomial.

(We'll see, e.g. $O(2^{\sqrt{\log_2 p}})$ subexponential)