Number Theory

Prime numbers \implies Fermat's Thm \implies Euler's Thm \implies Miller-Rabin

\mod, \mod \text{ inverse (already seen)} \implies \text{ Primility Test}

\implies RSA requires the use of 2 large primes +
reason it works is based on Euler's Thm.

\* Discrete Log Problem (El Gamal Public Key System)

What is a discrete log? \( \log_2 64 = 6 \)

because \( 2^6 = 64 \)

\log \text{ is inverse of the exponent problem.}

Input (Given): Ans, Base
Output: Exponent

New Problem

Input: Ans, Base, Mod
Output: exponent

What's smallest positive int \( x \) that makes this true?

\[ 5^x \equiv 6 \pmod{11} \]
<table>
<thead>
<tr>
<th>Exp</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ans</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ 3^x \equiv 6 \pmod{11} \]

\[ 2^x \equiv 6 \pmod{11} \]

This is the discrete log problem for prime numbers, \( p \), we can calculate the cycle length for each possible base in modular exponentiation. We previously proved that the cycle length, \( c \), for any base \( b \) divides evenly into \( p-1 \).

\( p = 11, \ p - 1 = 10 \), all cycle lengths are either 1, 2, 5 or 10.
Did program for $p = 11$

\[ \begin{align*}
\text{len} & = 10 \\
\text{cycle len} & = 5 \\
3, 4, 5, 9 & \phi(5) = 4 \\
\text{cycle len} & = 2 \\
10 & \phi(2) = 1 \\
\text{cycle len} & = 1 \\
1 & \phi(1) = 1
\end{align*} \]

These values have property that they produce each possible non-zero mod value in a reasonably unpredictable order.

Once we know a primitive root exists, then we can prove there are exactly $\phi(p-1)$ of them.

\[ 2^{10} \equiv 1 \mod 11 \]

and \[ 2^2 \equiv 4 \mod 11 \]

\[ 4^5 \equiv (2^2)^5 \equiv 2^{10} \equiv 1 \mod 11 \]
The discrete log problem will be used with \( \text{mod} = \text{prime}, \text{ base} = \text{primitive root} \to \text{each possible ans exists.} \)

Easiest/Slowest alg to solve

\[
\begin{align*}
    & b, a, p \mod \\
    & \text{res} = 1 \\
    & \text{for } (\text{int } i = 0; i < p; i++) \text{ if } \text{(res} = = a) \text{ return } i; \\
    & \text{res} = (\text{res} + b) \mod p; \\
    & \text{run time } O(p + \text{operr})
\end{align*}
\]

An alg that is a little bit faster
let \( n = \lceil \sqrt{p} \rceil \text{ceilings (least int } \geq \sqrt{p}) \)
goal: find integers \( c \) and \( d \) such that

\[
\frac{nc - d}{a} = b \pmod{p}
\]

Imagine \( p = 97 \quad n = 10 \)

\[
\begin{array}{cccc}
    c=1 & d=0,1,2,3 & \ldots & 10 \\
    10-0 & 10-1 & 10-2 & \ldots & 10-10 \\
    a & a & a & \ldots & a \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
    \bar{a} & \bar{a} & \bar{a} & \ldots & \bar{a} \\
\end{array}
\]
1. Calculate $a^{nc}$ for $c = 0, 1, 2, \ldots n$, store answers in a chart.

2. Calculate $a^d$ for $d = 0, 1, 2, \ldots n$.

\[ a^d (a^{nc-d}) = (b)^d \mod p \]

\[ \Rightarrow a^{nc} = b \cdot a^d \mod p \]

<table>
<thead>
<tr>
<th>TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^c$</td>
</tr>
</tbody>
</table>

For each $d$ multiply $b \cdot a^d$ and see if $a^{nc}$ is in the lookup chart.

Step 1 takes $O(\sqrt{p})$ time.

Step 2 takes $O(\sqrt{p})$ time.

$O(\sqrt{p})$