Discrete Logarithm

Lucas

Shanghai Jiao Tong University

December 25, 2016

Problem

Find a smallest non-negative number x satisfying

$$A^x = B \ (mod \ P)$$

which P is a prime, $A, B \in [0, P)$.

Naive Approach

According to Fermat Theory, when A, P is coprime, we have:

$$A^P = A \ (mod \ P)$$

the only special case is $A = 0$, and in this case, B must be zero.

For $A \neq 0$, we can just iterate x from 0 to $P - 1$ checking if $A^x = B \ (mod \ P)$. the complexity is $O(P)$.

Yet Another Naive Algorithm

We mapped $A^x \to x, x \in [0, P-1)$ by using a Hash-Table.

For finding B, we just need to query B in this Hash-Table, which only requires $O(1)$ time.

Preparing the Hash-Table needs $O(P)$ time, which the total complexity is $O(P+1) = O(P)$.

Balanced Programming

- First we choose a number $S \in [1, P-1]$.
- We mapped $A^x \to x, x \in [0, S)$ by using a Hash-Table.
- Calculate A^{-S} by using Fast Exponentiation.

In this step, we needs $S + \log P$ operations.

Balanced Programming

x can be represented as $i \times S + j$, we can transform the equation:

$$A^{i \times S + j} = B \Leftrightarrow A^j = B \times (A^{-S})^i \ (mod \ P)$$

which $j \in [0, S)$, and $i \leq \frac{P}{S}$.

We can just iterate i from 0 to $\frac{P}{S}$ checking if $B \times (A^{-S})^i$ is in Hash-Table. In worst occasion, we need to iterator $\frac{P}{S} + 1$ times.

Total complexity evaluation

As you can see, the total operations we need to do is

$$S + \log P + \frac{P}{S} + 1$$

we want to choose S optimally to have the best total complexity.

when $S = \sqrt{P}$, the total operations is:

$$S + \log P + \frac{P}{S} + 1 = 2\sqrt{P} + \log P + 1 = O(\sqrt{P})$$

thus we get a $O(\sqrt{P})$ algorithm by choosing S optimally.