幂运算

问题描述：计算$g^A(mod\ N)$

直接方法：反复不断的乘g，直到得到g^A

$$g_1 \equiv g\ (mod\ N),\ g_2 \equiv g \cdot g_1\ (mod\ N),\ g_3 \equiv g \cdot g_2\ (mod\ N),$$

$$g_4 \equiv g \cdot g_3\ (mod\ N),\ g_5 \equiv g \cdot g_4\ (mod\ N), ...$$

需要进行A次乘法运算，如果A很大，算法将不再实用。
例如$A \approx 2^{1000}$，算法运行的时间将会比现在宇宙的年龄更长

幂运算

快速幂算法：

1. 计算A 的二进制表达形式：

$$A = A_0 \cdot 2^0 + A_1 \cdot 2^1 + A_2 \cdot 2^2 + \cdots + A_r \cdot 2^r$$

$$A_0, A_1, \ldots, A_r \in \{0, 1\}$$

A_r 表示二进制数A 的最高位

2. 通过不断的平方，得到g^{2^i} ：

$$\begin{aligned}
a_0 &\equiv g && (mod\ N)\\
a_1 &\equiv a_0^2 \equiv g^2 && (mod\ N)\\
a_2 &\equiv a_1^2 \equiv g^{2^2} && (mod\ N)\\
a_3 &\equiv a_2^2 \equiv g^{2^3} && (mod\ N)\\
&\vdots \qquad \vdots && \vdots\\
a_r &\equiv a_{r-1}^2 \equiv g^{2^r} && (mod\ N)
\end{aligned}$$

幂运算

3. 通过以下方式计算 $g^A(mod\ N)$：

$$
\begin{aligned}
g^A &= g^{A_0\cdot 2^0+A_1\cdot 2^1+A_2\cdot 2^2+\cdots+A_r\cdot 2^r} \\
&= g^{A_0}\cdot(g^2)^{A_1}\cdot(g^{2^2})^{A_2}\cdot(g^{2^3})^{A_3}\cdots(g^{2^r})^{A_r} \\
&= a_0^{A_0}\cdot a_1^{A_1}\cdot a_2^{A_2}\cdot a_3^{A_3}\cdots a_r^{A_r}\ \ (mod\ N)
\end{aligned}
$$

注意：$a_0, a_1, \ldots, a_r$ 已经在步骤2中经过r次乘法运算得出。

步骤3也只需要进行r次乘法运算。

$r \approx \log_2 A$

当 $A \approx 2^{1000}$ 时，最多进行2000次乘法运算