# 高级算法设计与分析 Lecture 5

授课时间: 2020 年 3 月 16 日 授课教师: 孙晓明 记录人: 陈祖志

## 1 More on Chernoff's Bound

### 1.1 Remark on Chernoff's Bound for $\delta \geq 1$

**定理 1** (Chernoff's Bound). $X_1, \ldots, X_n$ 为独立的 $0-1$ 随机变量，$\Pr(X_i = 1) = p_i$，$\Pr(X_i = 0) = 1 - p_i$。记 $X = X_1 + \cdots + X_n$，$\mathbb{E}(X) = p_1 + \cdots + p_n = \mu$，则有

$$\Pr(X \geq (1+\delta)\mu) \leq \left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^\mu, \forall \delta \geq 0 \tag{1.1}$$

$$\Pr(X \leq (1-\delta)\mu) \leq \left[\frac{e^\delta}{(1-\delta)^{1-\delta}}\right]^\mu, 0 \leq \delta \leq 1 \tag{1.2}$$

*以下的界比较松但是更加直观:*

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2 \mu}{3}}, 0 \leq \delta \leq 1 \tag{1.3}$$

$$\Pr(X \leq (1-\delta)\mu) \leq e^{-\frac{\delta^2 \mu}{2}}, 0 \leq \delta < 1 \tag{1.4}$$

*结合式 1.1和式 1.4，可得以下形式:*

$$\Pr(|X - \mu| \geq \delta\mu) \leq 2e^{-\frac{\delta^2 \mu}{3}}, 0 \leq \delta < 1 \tag{1.5}$$

*当 $\delta \geq 1$,*

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta \mu}{3}}, \forall \delta \geq 1 \tag{1.6}$$

**证明** 仅证明式 (1.6)，根据 (1.1) 仅需考虑下列不等式（constant $c$ 待定）:

$$\left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^\mu \leq e^{-c\delta\mu}, \forall\ \delta \geq 1 \tag{1.7}$$

不等式两边取对数，整理后可得:

$$c \leq \frac{(1+\delta)\ln(1+\delta) - \delta}{\delta}, \forall \delta \geq 1$$

令

$$f(x) = \frac{(1+\delta)\ln(1+\delta) - \delta}{\delta}, \forall \delta \geq 1$$

对 $f(x)$ 求导后可得

$$f^{'}(x) = \frac{\delta - \ln(1+\delta)}{\delta^2}, \forall \delta \geq 1$$

不难验证

$$f^{'}(x) \geq 0, \forall \delta \geq 1$$

因此

$$f(x) \geq f(1) = 2\ln(2) - 1 \approx 0.3863, \forall \delta \geq 1$$

欲使式 1.7成立，$c$ 只需要满足下列条件即可:

$$c \leq 2\ln(2) - 1 \approx 0.3863$$

显然，$c = \frac{1}{3}$ 是满足的。 □

### 1.2 Random Sampling Revisit

回顾随机采样问题，在离散集合 $U$ 中有某个子集 $T$，欲估计 $p = \frac{|T|}{|U|}$ 的大小。为此，在 $U$ 上进行 $n$ 次的独立均匀随机采样，定义随机变量 $X_i$:

$$X_i = \begin{cases} 0, & the\ i-th\ sample\ \notin T \\ 1, & otherwise \end{cases}$$

定义 $\hat{p} = \frac{1}{n}\sum_{i=1}^{n} X_i$。给定参数 $\epsilon, \delta > 0$，现在对估计值的相对误差有如下要求:

$$\Pr(\hat{p} \in [(1-\delta)p, (1+\delta)p]) \geq 1 - \epsilon$$

用 Chebyshev Inequality 估计, 可以得出采样数 $n \geq \frac{1}{\delta^2 p\epsilon}$。下面用 Chernoff's Bound 给出一个更紧的界:

$$\begin{aligned} & \Pr(\hat{p} \in [(1-\delta)p, (1+\delta)p]) \\ =& \Pr(|\hat{p} - p| \leq p\delta) \\ =& \Pr(|n\hat{p} - np| \leq np\delta) \\ =& \Pr(|X - \mu| \leq \mu\delta) \\ =& 1 - \Pr(|X - \mu| > \mu\delta) \\ \geq& 1 - 2e^{-\frac{\mu\delta^2}{3}} \\ =& 1 - 2e^{-\frac{np\delta^2}{3}} \\ \geq& 1 - \epsilon \end{aligned}$$

由此可得 $n \geq \frac{3\ln\frac{2}{\epsilon}}{p\delta^2}$。Chernoff's Bound 计算出来的界为 $\ln\frac{1}{\epsilon}$ 量级的，而 Chebyshev Inequality 计算出的界为 $\frac{1}{\epsilon}$ 量级，当 $\epsilon$ 足够小时，二者是指数级别的差距。

## 2 Preliminary on Number Theory

**素性检验** 给定一个大于 1 的整数 $N$，判断 $N$ 是否为素数。

由于数据在计算机使用二进制存储的，整数 $N$ 可表示为 $(b_n \dots b_1)_2$。因此，在分析一个素性检验算法时，算法输入的大小应该是 $O(\log_2 N)$ 而非 $O(N)$。

一个平凡的素性检验算法，依次测试 $N$ 是否能被 $2, 3, \dots, \sqrt{N}$ 整除。该算法的时间复杂度是 $O(2^{\frac{1}{2}\log N})$，是指数时间复杂度的。

### 2.1 Euclidean Algorithms

欧几里得算法可用于计算两个整数 $a, b$ 的最大公因数，该算法迭代进行带余除法计算，计算过程如下：

$$\begin{aligned} a &= q_1 b + r_1, \quad 0 < r_1 < b \\ b &= q_2 r_1 + r_2, \quad 0 < r_2 < r_1 \\ r_1 &= q_3 r_2 + r_3, \quad 0 < r_3 < r_2 \\ &\cdots\cdots \\ r_{n-2} &= q_n r_n - 1 + r_n, \quad 0 < r_n < r_{n-1} \\ r_{n-1} &= q_{n+1} r_n \end{aligned} \tag{2.1}$$

最终输出 $r_n$ 即为 $a, b$ 的最大公因数。该算法为多项式时间复杂度。

**引理 2.** *若整数 $a, b$ 互质，则同余方程*

$$ax \equiv 1 \pmod b \tag{2.2}$$

*有解，且可以使用 Euclidean 算法求解。*

**证明** 根据式 (2.1)，$r_1, r_2, \ldots, r_n$ 可以改写为以下形式

$$\begin{aligned} r_1 =& a + q_1 b \\ r_2 =& b + q_2 r_1 \\ &\cdots \\ r_n =& r_{n-2} + q_n r_{n-1} \end{aligned}$$

将上式等式右端的 $r_i$ 逐一替换为 $a$ 与 $b$ 的线性组合，可得

$$\begin{aligned} r_1 =& a + q_1 b \\ r_2 =& c_2 a + d_2 b \\ &\cdots \\ r_n =& c_n a + d_n b \end{aligned}$$

由 $r_n = \gcd(a, b) = 1$ 可得:

$$1 = c_n a + d_n b$$

$c_n$ 为同余方程 2.2的解。 □

### 2.2 Fermat's Little Theorem

**引理 3.** *$p$ 为质数，整数 $a$ 满足 $gcd(a, p) = 1$，对于任意的 $i, j \in \{1, 2, \ldots, p-1\}$，若 $i \neq j$，则 $ia(mod\ p) \neq ja(mod\ p)$*

**证明** 反设 $ia(mod\ p) = ja(mod\ p)$, 则 $p|(i-j)a$。又 $gcd(a, p) = 1$，因此 $p|(i-j)$。由 $i, j$ 的取值区间可知 $-(p-1) < i - j < p - 1$，因此 $i = j$，矛盾。假设不成立。 □

**定理 4** (费马小定理). $p$ 为质数，对于任意的正整数 $a$，若 $p \nmid a$，则 $a^{p-1} \equiv 1(mod\ p)$

**证明** 考虑集合 $\{a(mod\ p), 2a(mod\ p), \ldots, (p-1)a(mod\ p)\}$, 由引理 3可知，该集合由 $p-1$ 个互不相同的元素组成。而该集合中元素可能的取值只有 $p-1$ 个。因此

$$\{a(mod\ p), 2a(mod\ p), \ldots, (p-1)a(mod\ p)\} = \{1, 2, \ldots, p-1\}$$

将集合中的元素相乘后取模，可得

$$\begin{aligned}
&a(2a)\ldots((p-1)a) \equiv (p-1)!(mod\ p) \\
\Rightarrow\ &p|(p-1)!(a^{p-1}-1) \\
\Rightarrow\ &p|(a^{p-1}-1) \\
\Rightarrow\ &a^{p-1} \equiv 1(mod\ p)
\end{aligned}$$

□

### 2.3 Fermat Primality Test

**费马素性检验** 给定一个整数 $N$, 需要判断 $N$ 是否为素数。则随机选取若干个 $a_i \in \{2, 3, \ldots, N-1\}$，判断

$$a_i^{N-1} \equiv 1(mod\ N) \tag{2.3}$$

是否成立。如果均成立，则判断 $N$ 为素数；否则，判断 $N$ 为合数。

根据费马小定理，如果 $N$ 为素数，则选取的任意 $a_i$，式 2.3均成立，一定可以通过费马素性检验；然而，如果 $N$ 为合数，$N$ 仍然有可能可以通过绝大部分的费马素性检验，比较极端的一个例子便是 Carmichael Number 。

**定义 5** (Carmichael Number). $N$ 为合数，对于任意的正整数 $a$，若 $gcd(a, N) = 1$，则 $a^{N-1} \equiv 1(mod\ N)$ 成立。称 $N$ 为 *Carmichael Number* 。

**例 1** 561 is a Carmichael number。

**证明** 561=3×11×17，对于任意的正整数 $a$，若 $gcd(a, N) = 1$，则必有

$$\begin{aligned}
gcd(a, 3) &= 1 \\
gcd(a, 11) &= 1 \\
gcd(a, 17) &= 1
\end{aligned}$$

由费马小定理可得

$$\begin{aligned}
a^2 &\equiv 1(mod\ 3) \\
a^{10} &\equiv 1(mod\ 11) \\
a^{16} &\equiv 1(mod\ 17)
\end{aligned} \tag{2.4}$$

由式 2.4 可得

$$
\begin{aligned}
a^{560} &\equiv (a^2)^{280} \equiv 1 (mod\ 3) \\
a^{560} &\equiv (a^{10})^{56} \equiv 1 (mod\ 11) \\
a^{560} &\equiv (a^{16})^{35} \equiv 1 (mod\ 17)
\end{aligned}
$$

从而有 $a^{560} \equiv 1(mod\ 3 \times 11 \times 17)$ □

## 2.4 Chinese Remainder Theorem

**定理 6.** 整数 $m_1, m_2, \ldots, m_t$ 两两互质，对于任意的整数 $n_i, i \in \{1, 2, \ldots, t\}$，同余方程组

$$
\begin{cases}
x \equiv n_1 \quad (mod\ m_1) \\
x \equiv n_2 \quad (mod\ m_2) \\
\qquad \vdots \\
x \equiv n_t \quad (mod\ m_t)
\end{cases}
\tag{2.5}
$$

在模 $M$ 意义下存在唯一解,$x = \sum_{i=1}^{m} n_i p_i M_i (mod\ M)$。其中 $M = m_1 m_2 \ldots m_t$,$\forall i \in \{1, 2, \ldots, t\}$,$M_i = \frac{M}{m_i}$,$p_i M_i \equiv 1(mod\ m_i)$。

**证明** 存在性：由 $m_1, m_2, \ldots, m_t$ 两两互质可知，$M_i$ 与 $m_i$ 互质，由引理 2，方程

$$M_i x \equiv 1(mod\ m_i)$$

有解，记为 $p_i$。

令 $x = \sum_{i=1}^{m} n_i p_i M_i$，下证 $x$ 为同余方程组 2.5的解。对于任意的 $i, j \in \{1, 2, \ldots, t\}$，若 $i \neq j$，由 $M_i$ 定义，有 $m_j | M_i$。因此

$$x \equiv n_j p_j M_j (mod\ m_j)$$

由 $p_j$ 的定义 $p_j M_j \equiv 1(mod\ m_j)$,

$$x \equiv n_j p_j M_j \equiv n_j (mod\ m_j)$$

因此 $x$ 为同余方程组 2.5的解。

唯一性：设 $x_1, x_2$ 均为同余方程组 2.5的解，则有

$$
\begin{cases}
m_1 | (x_1 - x_2) \\
m_2 | (x_1 - x_2) \\
\qquad \vdots \\
m_t | (x_1 - x_2)
\end{cases}
$$

由于 $\{m_1, \ldots, m_t\}$ 两两互质，因此 $m_1 m_2 \cdots m_t | (x_1 - x_2)$，即 $x_1 \equiv x_2 (mod\ M)$。 □

# 3 Prime Number Set is in BPP

记

$$Prime = \{2, 3, 5, \dots\}$$
$$Composite = \overline{Prime} = \{4, 6, 8, \dots\}$$

以下算法表明

$$Prime \in BPP$$

## 3.1 A Two-sided Error Algorithm

**Algorithm 1** A two sided error algorithm for primality test

**Input:** A odd integer $N \geq 2$

**Output:** prime or composite

**if** $\exists a, b \in \mathbb{N}, a > 1$ s.t. $N = a^b$ **then** ▷ 如果 $N$ 可以写成 $a^b$，则判断其为合数
    **return** composite
**end if**
Randomly and independently choose $a_1, a_2 \in \{2, 3, \dots, N-1\}$
**if** $gcd(a_1, N) > 1$ or $gcd(a_2, N) > 1$ **then** ▷ 如果 $N$ 与 $a_1$,$a_2$ 有公因数，判断其为合数
    **return** composite
    **if** ${a_1}^{N-1} (mod\ N) \neq 1$ or ${a_2}^{N-1} (mod\ N) \neq 1$ **then** ▷ 费马素性检验
        **return** composite
        **if** ${a_1}^{\frac{N-1}{2}} (mod\ N) \neq \pm 1$ or ${a_2}^{\frac{N-1}{2}} (mod\ N) \neq \pm 1$ **then**
            **return** composite
            **if** ${a_1}^{\frac{N-1}{2}} (mod\ N) = 1$ and ${a_2}^{\frac{N-1}{2}} (mod\ N) = 1$ **then**
                **return** composite
            **else** ▷ 仅在此处会判断其为素数
                **return** prime
            **end if**
        **end if**
    **end if**
**end if**

## 3.2 Time Complexity Analysis

**引理 7.** 给定正整数 $a, b$，可在 $O(\max\{\log^3 a, \log^3 b\})$ 的多项式时间内计算 $a^b$。

**证明** 依次计算 $a^2, a^4, \dots, a^{2^{\lfloor \log_2 b \rfloor}}, a^b$。总共需要 $O(\log b)$ 次乘法运算，每次乘法运算可在 $O(\log^2 a)$ 时间内完成。因此计算 $a^b$ 可在 $O(\log^2 a \log b) \leq O(\max\{\log^3 a, \log^3 b\})$ 时间内完成。 □

**引理 8.** 给定正整数 $N > 1$，可在多项式时间内判断是否存在 $a, b \in \mathbb{N}, a > 1, s.t. N = a^b$

**证明** 若存在这样的 $a,b$, 则 $b \leq \lceil \log_2 N \rceil$。对于每一个可能的 $b$ 的取值, 对区间 $\{2,3,\ldots,N-1\}$ 用二分法查找 $a$。如果这样的 $a,b$ 存在，通过该方法一定能找到。该方法需要 $O(\log^2 N)$ 次幂运算，由引理 7可知，每次幂运算只需要多项式时间，因此，该方法是多项式时间复杂度。 □

**定理 9.** *Algorithm 1为多项式时间复杂度算法。*

**证明** 由引理 8，算法 1的第 1行至第 3行可在多项式时间内完成。由于欧几里得算法为多项式时间复杂度算法，因此算法 1的第 5行也可在多项式时间内完成。由引理 7算法 1的第 7行至第 18行中的幂运算也可在多项式时间内完成。因此，Algorithm 1为多项式时间复杂度算法。 □

### 3.3 Correctness Analysis

**引理 10.** *若 $p$ 为奇质数，$gcd(a,p)=1$，则 $a^{\frac{p-1}{2}}(mod\ p) \in \{1,-1\}$*

**证明** 由费马小定理，

$$\begin{aligned}
& a^{p-1} \equiv 1(mod\ p) \\
\Rightarrow\ & p|(a^{p-1}-1) \\
\Rightarrow\ & p|(a^{\frac{p-1}{2}}-1)(a^{\frac{p-1}{2}}+1) \\
\Rightarrow\ & p|(a^{\frac{p-1}{2}}-1)\ or\ p|(a^{\frac{p-1}{2}}+1) \\
\Rightarrow\ & a^{\frac{p-1}{2}} \equiv 1(mod\ p)\ or\ a^{\frac{p-1}{2}} \equiv -1(mod\ p)
\end{aligned}$$

□

**引理 11.** *整数 $p$ 为奇质数，令*

$$\begin{aligned}
A &= \{a|a^{\frac{p-1}{2}} \equiv 1(mod\ p), a \in \{1,2,\ldots,p-1\}\} \\
B &= \{a|a^{\frac{p-1}{2}} \equiv -1(mod\ p), a \in \{1,2,\ldots,p-1\}\}
\end{aligned}$$

*则集合 $A$ 中的元素个数与集合 $B$ 中的元素个数相等，即 $|A|=|B|$。*

**证明** 由引理 10，$\forall a \in \{1,2,\ldots,p-1\}$，$a \in A$ or $a \in B$，因此 $|A|+|B|=p-1$。
由于 $p$ 为质数，$\mathbb{Z}_p$ 上的 $n$ 次同余方程至多有 $n$ 个不同的根。因此 $|A| \leq \frac{p-1}{2}$，$|B| \leq \frac{p-1}{2}$。
可得以下关系：$\frac{p-1}{2} \geq |A| = p-1-|B| \geq p-1-\frac{p-1}{2} = \frac{p-1}{2}$。因此 $|A|=|B|=\frac{p-1}{2}$。 □

**引理 12.** *$N$ 是奇合数，且不能写成 $N=a^b$(其中 $a,b$ 是正整数)。正整数 $c,d$ 满足 $\gcd(d,N)=1$，且 $d^c \equiv -1 \pmod N$。令*

$$\begin{aligned}
A &= \{k|k^c(mod\ N) \neq \pm 1, k \in \{2,\ldots,N-1\}\}, \\
B &= \{k|k^c(mod\ N) = \pm 1, k \in \{2,\ldots,N-1\}\}.
\end{aligned}$$

*则集合 $A$ 的元素个数不少于集合 $B$ 的元素个数，即 $|A| \geq |B|$。*

**证明** 证明分为三步，step 1：证明 $A \neq \phi$.(i.e. $\exists z_0 \in A$)；step 2：证明 $\forall b \in B, bz_0 \in A$；step 3：$\forall b_1,b_2 \in B, if\ b_1 \neq b_2, then\ z_0b_1(mod\ N) \neq z_0b_2(mod\ N)$。

Step 1：$N$ 为合数且 $N$ 含有两个不同的质因子，因此 $N = pq, gcd(p, q) = 1$。考虑同于方程组

$$\begin{cases} z \equiv d \quad (mod\ p) \\ z \equiv 1 \quad (mod\ q) \end{cases} \tag{3.1}$$

由中国剩余定理，同余方程组 3.1存在唯一解，记为 $z_0$，则

$$\begin{cases} {z_0}^c \equiv d^c \quad (mod\ p) \\ {z_0}^c \equiv 1 \quad (mod\ q) \end{cases}$$

又由 $d^c \equiv -1(mod\ N)$

$$\begin{cases} {z_0}^c \equiv -1 \quad (mod\ p) \\ {z_0}^c \equiv 1 \quad (mod\ q) \end{cases}$$

若 ${z_0}^c(mod\ N) = \pm 1$，则可推出 $p|2$ 或 $q|2$，这与 $N$ 是奇合数矛盾。因此 $z_0(mod\ N) \in A$。

Step 2：$\forall b \in B$，$(z_0 b)^c \equiv \pm {z_0}^c(mod\ N)$，因此 $(z_0 b)^c(modN) \neq \pm 1$。

Step 3：由 $gcd(d, N) = 1$，不难证明 $gcd(z_0, N) = 1$。$\forall b_1, b_2 \in B$，若 $z_0 b_1 \equiv z_0 b_2(mod\ N)$，则 $N|z_0(b_1 - b_2)$。由 $gcd(z_0, N) = 1$，则必有 $N|(b_1 - b_2)$。由 $b_1, b_2$ 的取值范围可知，$b_1 = b_2$。 □

**定理 13.** $Prime \in BPP$

**证明** 借助 Algorithm 1来证明，由定理 9可知，Algorithm 1为多项式时间复杂度。因此，只需证明：若 $N$ 是素数，则 Algorithm 1输出 *composite* 的概率小于 $\frac{1}{3}$；若 $N$ 是合数，则 Algorithm 1输出 *prime* 的概率小于 $\frac{1}{3}$。

$N$ **为素数**：根据素数的性质，费马小定理以及引理 10可知，Algorithm 1的第 1行至第 9行均可以通过。出错的情况只能是 ${a_1}^{\frac{N-1}{2}}(mod\ N) = 1$ 且 ${a_2}^{\frac{N-1}{2}}(mod\ N) = 1$。由引理 11，$\Pr({a_1}^{\frac{p-1}{2}} \equiv 1(mod\ N)) = \frac{1}{2}$，$\Pr({a_2}^{\frac{p-1}{2}} \equiv 1(mod\ N)) = \frac{1}{2}$，因此

$$\begin{aligned}
&\Pr(\text{N is prime and Algorithm 1 output composite}) \\
=&\Pr({a_1}^{\frac{p-1}{2}} \equiv 1(mod\ N) \cap {a_2}^{\frac{p-1}{2}} \equiv 1(mod\ N)) \\
=&\Pr({a_1}^{\frac{p-1}{2}} \equiv 1(mod\ N)) \Pr({a_2}^{\frac{p-1}{2}} \equiv 1(mod\ N)) \\
=&\frac{1}{2} \times \frac{1}{2} \\
<&\frac{1}{3}
\end{aligned}$$

**N 为合数**：若存在 $a > 1, b$ 使得 $N = a^b$，则该算法一定不会出错。令 $c = \frac{N-1}{2}$, 若不存在正整数 $d$，使得 $gcd(d, N) = 1$，$d^c \equiv -1(mod\ N)$，则该算法也一定不会出错。因此，仅考虑 $N \neq a^b$ 且存在正整数 $d$，使得 $gcd(d, N) = 1$，$d^c \equiv -1(mod\ N)$，根据引理 12，

$$\begin{aligned}
&|\{k|k^c(mod\ N) \neq \pm 1, k \in \{2, \ldots, N-1\}\}| \\
\geq&|\{k|k^c(mod\ N) = \pm 1, k \in \{2, \ldots, N-1\}\}|
\end{aligned}$$

即任取 $a_i \in \{2,3,\ldots,N-1\}$，则 $\Pr(a_i^{\frac{N-1}{2}} \equiv \pm1(mod\ N)) \leq \frac{1}{2}$

$$\begin{aligned}
&\Pr(N \text{ is composite and Algorithm 1 output prime})\\
=&\Pr\Big(\ gcd(a_1,N)=1 \cap gcd(a_2,N)=1\\
&\quad \cap a_1{}^{N-1} \equiv 1(mod\ N) \cap a_2{}^{N-1} \equiv 1(mod\ N)\\
&\quad \cap a_1{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N) \cap a_2{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N)\\
&\quad \cap (a_1{}^{\frac{N-1}{2}} \equiv 1(mod\ N) \cup a_2{}^{\frac{N-1}{2}} \equiv 1(mod\ N))\ \Big)\\
\leq&\Pr(a_1{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N) \cap a_2{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N))\\
=&\Pr(a_1{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N) Pr(a_2{}^{\frac{N-1}{2}} \equiv \pm1(mod\ N)\\
\leq&\frac{1}{2}\times\frac{1}{2}\\
<&\frac{1}{3}
\end{aligned}$$

□

# 4 Composite Number Set is in RP

## 4.1 Miller-Rabin Algorithm

**Algorithm 2** Miller-Rabin Algorithm

**Input:** A odd integer $N \geq 2$

**Output:** prime or composite

**if** $\exists a,b \in \mathbb{N}, a>1$ s.t. $N = a^b$ **then** ▷ 如果 $N$ 可以写成 $a^b$，则判断其为合数
  **return** composite
**end if**
Randomly choose $a \in \{2,3,\ldots,N-1\}$
**if** $a^{N-1}(mod\ N) \neq 1$ **then** ▷ 费马素性检验
  **return** composite
**end if**
Set $m := N-1$
**while** $m \equiv 0(mod\ 2)$ **do** ▷ $N-1 = 2^s t, t$ 为奇数。逐一检查 $2^{s-1}t$,...,$2^1t$,$2^0t$
  $m := \frac{m}{2}$
  **if** $a^m(mod\ N) \neq \pm1$ **then**
    **return** composite
  **end if**
  **if** $a^m(mod\ N) = -1$ **then**
    **return** prime
  **end if**
**end while**
**return** prime

### 4.2 Time Complexity Analysis

**定理 14.** *Algorithm* 2 为多项式时间复杂度算法。

**证明** 由引理 8，算法第 1行至第 3行可在多项式时间内完成。算法第 5行至第 17行需要进行 $O(\log_2 N)$ 次幂指数运算，由引理 7，也可在多项式时间内完成。 □

### 4.3 Correctness Analysis

**定理 15.** $Composite \in RP$

**证明** 借助 *Algorithm* 2 证明。只需证明：若 $N$ 为素数，则 *Algorithm* 2 输出 *prime*；若 $N$ 为合数，则 *Algorithm* 2 输出 *prime* 的概率不超过 $\frac{1}{2}$。

$N$ **为素数**：由素数定义以及费马小定理，$N$ 必能通过 *Algorithm* 2 的第 1 行和第 5 行。假设算法输出 *composite*，那么只能是算法的第 12 行。此时有 $a^k(mod\ N) \neq \pm 1$ 以及 $a^{2k}(mod\ N) = 1$。由 $a^{2k}(mod\ N) = 1$ 以及 $N$ 为素数的性质，可以推出 $a^k(mod\ N) \in \{-1, 1\}$，矛盾。因此，*Algorithm* 2 不可能在第 12行返回 *composite*。在 $N$ 为素数的情况下，该算法总是返回 *prime*。

$N$ **为合数**：若存在 $a, b, a > 1$ 使得 $N = a^b$，则该算法一定会输出 *composite*。因此，考虑不存在 $a, b, a > 1$ 使得 $N = a^b$ 的情况。此时，如果算法出错，说明算法在第 15 行或第 18 行输出 *prime*，分两种情况讨论：

**Case 1**：*Algorithm* 2 在第 15 行输出 *prime*。记此时的 $m = 2^{k-1}t$，令

$$A_{k-1} = \{a | a^{2^{k-1}t}(mod\ N) \neq \pm 1, k \in \{2, \ldots, N-1\}\}$$
$$B_{k-1} = \{a | a^{2^{k-1}t}(mod\ N) = \pm 1, k \in \{2, \ldots, N-1\}\}$$

由引理 12 可知，$|A_{k-1}| \geq |B_{k-1}|$。

**Case 2**：*Algorithm* 2 在第 18 行输出 *prime*。此时 $m = 2^0 t$，$m$ 为奇数。又 $gcd(N-1, N) = 1, (N-1)^m \equiv -1(mod\ N)$。令

$$A_0 = \{a | a^t(mod\ N) \neq \pm 1, k \in \{2, \ldots, N-1\}\}$$
$$B_0 = \{a | a^t(mod\ N) = \pm 1, k \in \{2, \ldots, N-1\}\}$$

由引理 12 可知，$|A_0| \geq |B_0|$。

结合 (Case 1) 和 (Case 2)，

$$\Pr(N \text{ is a composite and } algorithm2 \text{ output prime}) \leq \frac{1}{2}$$

□

**Remark** 2002 年 Agrawal-Kayal-Saxena 提出了素数判定的一个确定性的多项式时间算法，i.e. $Prime \in P$。