高级算法设计与分析 Lecture 4

授课时间: 2021 年 3 月 29 日 授课教师: 孙晓明 记录人: 韩雨

1 最大负载问题的补充

Two-Choice Load Banlancing 回忆 X_i 表示第 i 个盒子中的小球数量。如果在每次投放时随机选择 2 个盒子，询问当前盒子内的球数，然后投放到比较少的一个盒子内，那么当 $m = n$ 时，以高概率 $\max X_i = \Theta(\ln\ln n)$[1]。

2 Chernoff's Bound

当 $m \sim \Theta(n\ln n)$ 时，以高概率有 $\forall X_i$，$X_i = \Theta(\frac{m}{n})$。为了证明这个结论，我们需要一个新的数学工具：Chernoff's Bound。

定理 1 (Chernoff's Bound)**.** 设 $X_1, \cdots, X_n$ 是独立的 $0-1$ 随机变量，满足 $\Pr(X_i = 1) = p_i, \Pr(X_i = 0) = 1 - p_i$。记 $X = X_1 + \cdots + X_n$，则 $\mathbb{E}(X) = p_1 + \cdots + p_n = \mu$，对 $\forall\ \delta \in (0, 1)$，有:

$$\Pr(X \geq (1+\delta)\mu) \leq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\mu} \leq e^{-\frac{\delta^2}{3}\mu} \tag{2.1}$$

$$\Pr(X \leq (1-\delta)\mu) \leq \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{\mu} \leq e^{-\frac{\delta^2}{2}\mu} \tag{2.2}$$

其中 (2.1) 式左半部分对 $\delta > 0$ 都成立。这里我们只给出 (2.1) 的证明。

证明 取 $\lambda > 0$:

$$\begin{aligned}
\Pr(X \geq (1+\delta)\mu) &= \Pr\left(e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right) \\
&\leq \frac{\mathbb{E}(e^{\lambda X})}{e^{\lambda(1+\delta)\mu}} && \text{Markov 不等式} \\
&= \frac{\mathbb{E}\left(e^{\lambda(X_1+\cdots+X_n)}\right)}{e^{\lambda(1+\delta)\mu}} \\
&= \frac{\mathbb{E}\left(e^{\lambda X_1}\right)\cdot\mathbb{E}\left(e^{\lambda X_2}\right)\ldots\mathbb{E}\left(e^{\lambda X_n}\right)}{e^{\lambda(1+\delta)\mu}} && X_i \text{ 相互独立} \\
&= \frac{[(1-p_1)+p_1e^{\lambda}]\cdot[(1-p_2)+p_2e^{\lambda}]\cdots[(1-p_n)+p_ne^{\lambda}]}{e^{\lambda(1+\delta)\mu}} \\
&= \frac{\left[1+p_1\left(e^{\lambda}-1\right)\right]\cdot\left[1+p_2\left(e^{\lambda}-1\right)\right]\cdots\left[1+p_n\left(e^{\lambda}-1\right)\right]}{e^{\lambda(1+\delta)\mu}} \\
&\leq \frac{e^{p_1\left(e^{\lambda}-1\right)}\cdot e^{p_2\left(e^{\lambda}-1\right)}\ldots e^{p_n\left(e^{\lambda}-1\right)}}{e^{\lambda(1+\delta)\mu}} && 1+x \leq e^x \\
&= \frac{e^{\left(e^{\lambda}-1\right)(p_1+\cdots+p_n)}}{e^{\lambda(1+\delta)\mu}} \\
&= \left[\frac{e^{e^{\lambda}-1}}{e^{\lambda(1+\delta)}}\right]^{\mu}
\end{aligned} \tag{2.3}$$

通过2.3式右边的结果求导可得，当 $\lambda = \ln(1+\delta)$ 时，该结果取最小值 $\left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\mu}$。

我们对 $\frac{e^\delta}{(1+\delta)^{1+\delta}}$ 取对数，得：

$$
\begin{aligned}
\ln\frac{e^\delta}{(1+\delta)^{1+\delta}} &= \delta-(1+\delta)\ln(1+\delta) \\
&= \delta-(1+\delta)\left(\delta-\frac{1}{2}\delta^2+\frac{1}{3}\delta^3-\cdots\right) \\
&= \delta-\left(\delta+\frac{1}{2}\delta^2-\frac{1}{6}\delta^3+\frac{1}{12}\delta^4-\cdots\right) \\
&= -\frac{1}{2}\delta^2+\frac{1}{6}\delta^3-\frac{1}{12}\delta^4+\cdots \\
&\le -\frac{1}{2}\delta^2+\frac{1}{6}\delta^3 && 0<\delta<1 \\
&\le -\frac{1}{2}\delta^2+\frac{1}{6}\delta^2 && 0<\delta<1 \\
&= -\frac{1}{3}\delta^2
\end{aligned}
$$

综上，我们有 $\Pr(X\ge(1+\delta)\mu)\le\left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^\mu\le e^{-\frac{\delta^2}{3}\mu}$。 □

作业. 证明 Chernoff's Bound 的另一式

有了这个工具之后，我们可以证明以下定理：

定理 2. 当 $m\ge 8n\ln n$ 时，$\Pr\left(\forall X_i,\frac{m}{2n}\le X_i\le\frac{2m}{n}\right)=1-o(1)$。

证明

$$
\begin{aligned}
\Pr\left(\forall X_i,\frac{m}{2n}\le X_i\le\frac{2m}{n}\right) &= 1-\Pr\left(\left(\max X_i>\frac{2m}{n}\right)\vee\left(\min X_i<\frac{m}{2n}\right)\right) \\
&\le 1-\Pr\left(\max X_i>\frac{2m}{n}\right)-\Pr\left(\min X_i<\frac{m}{2n}\right) && \text{Union Bound}
\end{aligned}
$$

1) 先证明 $\Pr\left(\max X_i>\frac{2m}{n}\right)=o(1)$。

$$
\begin{aligned}
\Pr\left(\max X_i>\frac{2m}{n}\right) &= \Pr\left(\left(X_1>\frac{2m}{n}\right)\vee\left(X_2>\frac{2m}{n}\right)\vee\cdots\vee\left(X_n>\frac{2m}{n}\right)\right) \\
&\le\sum_{i=1}^{n}\Pr\left(X_i>\frac{2m}{n}\right) && \text{Union Bound} \\
&= n\Pr\left(X_1>\frac{2m}{n}\right) && \text{这些概率同分布}
\end{aligned}
$$

我们定义 $Y_j(1\le j\le m)$，表示第 j 个球是否投放进第 1 个盒子中，他们都是独立的 $0-1$ 随机变量：

$$
Y_j=\begin{cases}1 & \text{if Ball\#j} \to \text{Bin\#1}\\ 0 & \text{otherwise}\end{cases}
$$

则 $X_1=\sum_{j=1}^m Y_j$, $\mathbb{E}(X_1)=\sum_{j=1}^m\mathbb{E}(Y_j)=\frac{m}{n}$。

由于 $X = \sum_i Y_i$，Y_i 是独立的伯努利变量，根据 Chernoff's bound，我们有

$$\begin{aligned}
\Pr\left(X_1 > \frac{2m}{n}\right) &= \Pr(X_1 > (1+1)\mathbb{E}(X_1)) \\
&\le \left[\frac{e^1}{(1+1)^2}\right]^{\frac{m}{n}} \\
&= \left(\frac{e}{4}\right)^{8\ln n} \\
&\le \left(\frac{1}{e^2}\right)^{\ln n} \\
&\le \frac{1}{n^2}
\end{aligned}$$

则 $\Pr\left(\max X_i > \frac{2m}{n}\right) \le n \cdot \frac{1}{n^2} = \frac{1}{n} = o(1)$。

2) 再证明 $\Pr\left(\min X_i < \frac{m}{2n}\right) = o(1)$。

$$\begin{aligned}
\Pr\left(\min X_i < \frac{m}{2n}\right) &= \Pr\left(\left(X_1 < \frac{m}{2n}\right) \vee \left(X_2 < \frac{m}{2n}\right) \vee \cdots \vee \left(X_n < \frac{m}{2n}\right)\right) \\
&\le \sum_{i=1}^{n} \Pr\left(X_i < \frac{m}{2n}\right) && \text{Union Bound} \\
&= n \Pr\left(X_1 < \frac{m}{2n}\right) && \text{这些概率同分布} \\
&= n \Pr\left(X_1 < \left(1 - \frac{1}{2}\right)\mathbb{E}(X_1)\right) \\
&\le n \left[\frac{e^{-\frac{1}{2}}}{(1-\frac{1}{2})^{\frac{1}{2}}}\right]^{\frac{m}{n}} \\
&= n \cdot \left(\frac{2}{e}\right)^{\frac{m}{2n}} \\
&\le n \cdot \left(\frac{2}{e}\right)^{4\ln n} \\
&= n^{-3} \cdot 2^{4\ln n} \\
&= n^{-3} \cdot n^{\ln 16} \\
&< n^{-0.2} = o(1)
\end{aligned}$$

综上，当 $m > 8n\ln n$ 时，$\Pr\left(\forall X_i, \frac{m}{2n} \le X_i \le \frac{2m}{n}\right) = 1 - o(1)$。

□

3 素性检验（Primality Test）

素性检验 给定一个整数 N，判断 N 是否为素数，为方便起见，我们在以下讨论中**只考虑 $N > 2$ 的情况**。

在计算机中，整数 N 是由一个二进制串 $b_1 b_2 \cdots b_n (n = \lceil \log_2 N \rceil)$ 表示的，我们一般认为算法的输入规模是 $n = \lceil \log_2 N \rceil$ 而非 N。

有一个朴素的素性检验算法，即依次检查 N 是否能被 $2,3,4,\cdots,\lfloor\sqrt{N}\rfloor$ 整除，该算法时间复杂度为 $O\left(\sqrt{N}\right)=O\left(2^{\frac{1}{2}n}\right)$，是指数级别的。我们期望找到一个更好的素性检验算法。

3.1 费马小定理（Fermat's Little Theorem）

定理 3 (费马小定理)**.** *p 为素数，对于任意正整数 a，若 $p \nmid a$，则*

$$a^{p-1} \equiv 1 \pmod p \tag{3.1}$$

证明 考虑数的序列 $a \pmod p, 2a \pmod p, \cdots, (p-1)a \pmod p$。

若 $ia \pmod p \equiv ja \pmod p$，则 $p \mid (i-j)a$。由于 $\gcd(a,p)=1$，故而 $p \mid (i-j)$。又由于 $|i-j|<p-1$，故 $i=j$。所以该序列元素互不相同，因此有：

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

将两个集合中元素各自相乘并取模，可以得到：

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

则 $p \mid (p-1)!(a^{p-1}-1)$。

由 $\gcd(p,p-1)=1$，得到 $p \mid (a^{p-1}-1)$，这等价于 $a^{p-1}\equiv 1 \pmod p$。 □

3.2 Fermat Primality Test

费马素性检验 对于整数 N，要判断其是否为素数，则随机选取若干个 $a_i \in 2,3,\cdots,N-1$，判断下式是否成立：

$$a^{N-1} \equiv 1 \pmod N \tag{3.2}$$

如果均成立，则判断 N 为素数；否则 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3.2都成立。但若 N 为合数，N 仍有可能通过对大多数 a_i 的费马素性检验，一个较为极端的例子就是 Carmichael Number。

定义 4 (Carmichael Number)**.** *若 N 为合数，且对于任意正整数 a，如果有 $\gcd(a,N)=1$，则 $a^{N-1}\equiv 1 \pmod N$ 成立。称 N 为 Carmichael Number。*

例 1 561 是一个 Carmichael Number。

证明 由于 $561=3*11*17$，若 $\gcd(a,561)=1$，则必有

$$\gcd(a,3)=1$$
$$\gcd(a,11)=1$$
$$\gcd(a,17)=1$$

则由费马小定理3.1

$$\begin{aligned}a^2 &\equiv 1 \pmod 3\\ a^{10} &\equiv 1 \pmod{11}\\ a^{16} &\equiv 1 \pmod{17}\end{aligned}$$

则有

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

从而 $a^{560} \equiv 1 \pmod{3*11*17}$。 □

3.3 中国剩余定理（Chinese Remainder Theorem）

在上面的证明中，我们构建了一组特殊的同余方程组。在这里，我们给出一个更一般的结论：

定理 5 (Chinese Remainder Theorem)**.** 整数 $m_1, m_2, \ldots, m_t$ 两两互质，对于任意整数 $n_i(i \in \{1, 2, \ldots, t\})$，同余方程组：

$$\begin{cases} x \equiv n_1 & \pmod{m_1} \\ x \equiv n_2 & \pmod{m_2} \\ \quad\vdots \\ x \equiv n_t & \pmod{m_t} \end{cases}$$

在模 M 意义下存在唯一解 x，$M = m_1 m_2 \cdots m_t$，$x = \sum_{i=1}^{m} n_i p_i M_i \pmod M$。其中 $M_i = \frac{M}{m_i}$，p_i 是 M_i 在模 m_i 意义下的逆（$p_i M_i \equiv 1 \pmod{m_i}$）。

4 素性检验的一个 BPP 算法

对于输入的大于 2 的整数 N

1. 检验是否 $2 \mid N$，若能整除，则判断 N 为合数，算法结束
2. 检验是否存在整数 $M, d > 1$，有 $M^d = N$，若存在，则 N 为合数，算法结束，检验方式如下
   (a) 枚举 $d = 1, 2, \cdots, \lfloor \log_2 N \rfloor$
   (b) 对于枚举的 d，二分查找 M，检查是否有 $M^d = N$，运算 M^d 可使用快速幂算法。
3. 常数次 (设为 c 次) 独立、均匀地从 $\{1, 2, \cdots, N-1\}$ 中选取 $a_i(1 \le i \le c)$，同步对它们进行如下检查

(a) 若 $\exists a_i, \gcd(a_i, N) \neq 1$，则判断 N 为合数，算法结束

(b) 若 $\exists a_i, a_i^{N-1} \not\equiv 1 \pmod N$，则判断 N 为合数，算法结束

(c) 若 $\exists a_i, a_i^{\frac{N-1}{2}} \pmod N \notin \{\pm 1\}$，则判断 N 为合数，算法结束

(d) 若 $\exists a_i, a_i^{\frac{N-1}{2}} \equiv -1 \pmod N$，则判断 N 为素数，算法结束

4. 若经过了以上步骤，算法仍未结束，则判断 N 为合数。

证明 为证明这个算法是 BPP 的，我们对该算法的时间复杂度和错误率进行分析。

时间复杂度分析 注意到输入规模 $n = \lceil \log_2(N) \rceil$。

步骤 1 只需要检查 N 的最后一位，可以在 $O(1)$ 时间完成。

步骤 2 中，一共枚举 $\lfloor \log_2(N) \rfloor$ 个数，二分查找 M 的次数为 $O(\lceil \log_2(N) \rceil)$，而由于 $d \leq \lfloor \log_2 N \rfloor$，无论是否使用快速幂算法，总能在 n 的多项式时间内计算得到 M^d。故步骤二也只用 n 的多项式时间完成。

步骤 3 的运算可以使用欧几里得算法和快速幂算法，也能在 N 的对数时间，即 n 的多项式时间内完成。

综上，该算法的运行时间是 n 的多项式时间。

错误率分析 对 N 是素数和非素数的情况，分别进行错误率分析。

1) 若 N 是素数，又 $N > 2$，则 N 为奇素数。

则 N 显然可以通过算法步骤 1、2 和步骤 3 的 (a),(b) 两步的检验。

由平方差公式可以得到

$$
\begin{aligned}
& a_i^{N-1} - 1 = (a_i^{\frac{N-1}{2}} - 1)(a_i^{\frac{N-1}{2}} + 1) \\
\Rightarrow & N \mid (a_i^{\frac{N-1}{2}} - 1)(a_i^{\frac{N-1}{2}} + 1) && \text{费马小定理} \\
\Rightarrow & N \mid (a_i^{\frac{N-1}{2}} - 1) \ or \ N \mid (a_i^{\frac{N-1}{2}} + 1) \\
\Rightarrow & a_i^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}
\end{aligned}
$$

那么 $a_i^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}$，$N$ 可以通过步骤 3 中 (c) 的检验。

由于 N 是素数，$\mathbb{Z}_N$ 上的 m 次同余方程至多有 m 个不同的根。

故 $a_i^{\frac{N-1}{2}} \pmod N = \pm 1$ 各自至多有 $\frac{N-1}{2}$ 个根，但同时 a_i 有 $N-1$ 种取值，且 $a_i^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}$，故 $\{1, 2, \cdots, N-1\}$ 中有一半的取值使得 $a_i^{\frac{N-1}{2}} \pmod N = 1$，另一半的取值使得 $a_i^{\frac{N-1}{2}} \pmod N = -1$。

则 $\Pr(a_i^{\frac{N-1}{2}} \pmod N = 1) = \frac{1}{2}$

那么 c 次取值都没有在 (d) 步结束算法的概率等于 $\frac{1}{2^c}$，只需取 $c \geq 2$，就能使判断错误率 $1 - \frac{1}{2^c} \leq \frac{1}{3}$。

2) 若 N 是合数，若其通过第 1 步，第 2 步以及步骤 3 中 (a)、(b)、(c) 的检测（否则直接判断是合数），则 N 是至少有两个素因子的奇合数，且对于所有 a_i，都有:

$$
\begin{aligned}
& \gcd(a_i, N) = 1 \\
& a_i^{N-1} \equiv 1 \pmod N \\
& a_i^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}
\end{aligned}
$$

由于在 (d) 被错判为素数，说明存在 i_*，有

$$\gcd(a_{i_*}, N) = 1$$
$$a_{i_*}^{\frac{N-1}{2}} \equiv -1 \pmod N$$

我们令:

$$A = \{a \in \mathbb{Z}_N^* | a^{\frac{N-1}{2}} \pmod N \notin \{\pm 1\}\}$$
$$B = \{a \in \mathbb{Z}_N^* | a^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}\}$$

下面我们说明 $|A| \geq |B|$:

首先，我们证明集合 A 非空，由 N 至少有两个素因子，可设 $N = m_1 m_2$，且 $\gcd(m_1, m_2) = 1$，对 a_{i_*} 考虑同余方程组:

$$\begin{cases} x \equiv a_{i_*} & \pmod{m_1} \\ x \equiv 1 & \pmod{m_2} \end{cases}$$

根据中国剩余定理，该方程组存在解，设为 x^*，则有

$$\begin{cases} (x^*)^{\frac{N-1}{2}} \equiv a_{i_*}^{\frac{N-1}{2}} & \pmod{m_1} \\ (x^*)^{\frac{N-1}{2}} \equiv 1 & \pmod{m_2} \end{cases}$$

由 $a_{i_*}^{\frac{N-1}{2}} \equiv -1 \pmod N$，可得 $a_{i_*}^{\frac{N-1}{2}} \equiv -1 \pmod{m_1}$。故而:

$$\begin{cases} (x^*)^{\frac{N-1}{2}} \equiv -1 & \pmod{m_1} \\ (x^*)^{\frac{N-1}{2}} \equiv 1 & \pmod{m_2} \end{cases}$$

则 $(x^*)^{\frac{N-1}{2}} \pmod{m_1 m_2} \notin \{\pm 1\}$，否则代入上式可得 $m_1 \mid 2$ 或者 $m_2 \mid 2$，这与 N 为奇合数矛盾。

同时，由于 a_{i_*} 与 m_1 互素，且 $x^* \equiv a_{i_*} \pmod{m_1}$，得到 x^* 与 m_1 互素。又由 $x^* \equiv 1 \pmod{m_2}$，得到 x^* 与 m_2 互素。故而 x^* 与 N 互素。

综上，$x^* \in A$，$|A|$ 非空。

同时，$\forall b \in B$，$(x^* b)^{\frac{N-1}{2}} \equiv \pm (x^*)^{\frac{N-1}{2}} \not\equiv 1 \pmod N$，且 $\gcd(x^* b, N) = 1$，可得 $x^* b \in A$，则 $aB \subseteq A$。根据群论或数论知识，$\forall b_1, b_2 \in B$，若 $x^* b_1 \equiv x^* b_2 \pmod N$，则有 $b_1 \equiv b_2 \pmod N$，这保证了 $|aB| = |B|$。所以有 $|A| \geq |B|$。

那么

$$\begin{aligned} \Pr(N\text{被判定为素数}) &\leq \Pr\Big(\bigwedge_{i=1}^{c} \big(a_i^{\frac{N-1}{2}} \pmod N \in \{\pm 1\}, \text{且 } a_i \text{ 在其他步骤没有成功判定 } N \text{ 为合数}\big)\Big) \\ &\leq \frac{1}{2^c} \end{aligned}$$

则只需取 $c \geq 2$，就能使判断错误率小于 $1 - \frac{1}{2^c} \leq \frac{1}{3}$。

综上，只需要取 $c \geq 2$，则该算法在多项式时间上运行，且双边错误率都小于 $\frac{1}{3}$。

$\square$

5 检验合数的一个 RP 算法

对于输入的大于 2 的整数 N

1. 检验是否 $2 \mid N$，若能整除，则判断 N 为合数，算法结束
2. 检验是否存在整数 $M, d > 1$，有 $M^d = N$。若存在，则 N 为合数，算法结束，检验方式如下
   (a) 枚举 $d = 1, 2, \cdots, \lfloor \log_2 N \rfloor$
   (b) 对于枚举的 d，二分查找 M，检查是否有 $M^d = N$，运算 M^d 可使用快速幂算法。
3. 均匀地从 $\{1, 2, \cdots, N-1\}$ 中随机选取 a，对它进行如下检查
   (a) 若 $\gcd(a, N) \neq 1$，则判断 N 为合数，算法结束
   (b) 若 $a^{N-1} \not\equiv 1 \pmod N$，则判断 N 为合数，算法结束
   (c) 设 $N-1 = 2^s t$, t 为奇数，依次对 $m = \{2^{s-1}t, 2^{s-2}t, \cdots, 2^1 t, 2^0 t\}$ 进行检查
      i. 若 $a^m \pmod N \notin \{\pm 1\}$，则判断 N 为合数，算法结束
      ii. 若 $a^m \pmod N = -1$，则判断 N 为素数，算法结束
4. 若经过了以上步骤，算法仍未结束，则判断 N 为素数。

证明 为证明这个算法是 RP 的，我们对该算法的时间复杂度和错误率进行分析。

时间复杂度分析 注意到输入规模 $n = \lceil \log_2(N) \rceil$

在第 3 步的 (c) 步骤之前，与之前的算法步骤一致。

对于第 3 步的 (c) 步骤，至多枚举 $\lfloor \log_2 N \rfloor$ 个 m，计算 $a^m \pmod N$ 也能在 n 的多项式时间内完成。

综上，该算法的运行时间是 n 的多项式时间。

错误率分析 对 N 是素数和非素数的情况，分别进行错误率分析

1) 若 N 是素数，又 $N > 2$，则 N 为奇素数。

同之前的算法，N 能通过步骤 1、2 和步骤 3 的 (a),(b) 检验。若对于某个 m，有 $a^{2m} \pmod N = 1$，则由平方差公式，$a^m \pmod N \in \{\pm 1\}$。在步骤 (c) 中，若对于某个 m，$a^m \pmod N = -1$，则判断 N 为素数，否则继续检验下去，总不会出现 $a^m \pmod N \notin \{\pm 1\}$ 的情况，故而能成功判断 N 为素数。

2) 若 N 是合数，根据之前的分析，如果算法出错，说明算法在第 3 步的 (c) 出错，或者在第 4 步错判该数为素数，此时 N 是至少有两个素因子的奇合数，且对选取的 a，有 $\gcd(a, N) = 1$。

若在 $\{1, 2, \cdots, N-1\}$ 中，存在 a 使得算法第 3 步中 (c) 步骤判断 N 为素数，设其中能最早判定出 N 的 $m_* = 2^k t$，则 $\exists a_*, a_*^{2^k t} \pmod N = -1$，则类似前一个定理的证明，令:

$$A_k = \{a_* \in \mathbb{Z}_N^* | a_*^{2^k t} \pmod N \notin \{\pm 1\}\}$$
$$B_k = \{a_* \in \mathbb{Z}_N^* | a_*^{2^k t} \pmod N \in \{\pm 1\}\}$$

则有 $|A_k| \geq |B_k|$

则对于随机选取的 a，$\Pr(a \in A_k) \geq \Pr(a \in B_k)$。

特别的，对于 $k=0$，由于 $(N-1)^t \equiv -1 \pmod N$，$|A_0| \geq |B_0|$ 始终成立，所以对于没在步骤 (c) 中对 $k \geq 1$ 作出判定的 a，$\Pr(a \in A_0) \geq \Pr(a \in B_0) \geq \Pr(a\text{落入第 4 步})$。

综上，$\Pr(N\text{为合数，但算法输出素数}) \leq \frac{1}{2}$。

$\square$

参考文献

[1] Mitzenmacher M D. The Power of Two Choices in Randomized Load Balancing[D]. UNIVERSITY of CALIFORNIA at BERKELEY, 1996.