高级算法设计与分析 Lecture 3

授课时间: 2021 年 3 月 22 日　授课教师: 孙晓明　记录人: 万宗祺

1 随机复杂性类之间的关系

1.1 不同错误率的 $\mathcal{RP}$, co-$\mathcal{RP}$ 与 $\mathcal{BPP}$

在第二讲中，我们提到了 $\mathcal{RP}$, co-$\mathcal{RP}$ 与 $\mathcal{BPP}$ 的定义。在 $\mathcal{RP}$ 与 co-$\mathcal{RP}$ 中，我们让单边错误率不大于 $\frac{1}{2}$，在 $\mathcal{BPP}$ 中，我们让算法在 $x \in L$ 和 $x \notin L$ 时的错误率都不大于 $\frac{1}{3}$。而实际上，错误率的选择无关紧要 (但需要保证 $\mathcal{BPP}$ 中的错误率严格小于 $\frac{1}{2}$)，我们将证明这一点。

定义 1 ($\mathcal{RP}_\epsilon$)**.** *语言 $L \in \mathcal{RP}_\epsilon$ 是说，存在一个多项式时间的算法 A, 以一个实例 x 以及一个随机比特串 r 为输入，并输出 0 或 1。并且其需要满足:*

- 若 $x \in L$, 则 $\Pr(A(x,r)=1) \geq 1-\epsilon$
- 若 $x \notin L$, 则 $\Pr(A(x,r)=0) = 1$

定义 2 (co-$\mathcal{RP}_\epsilon$)**.** *语言 $L \in$ co-$\mathcal{RP}_\epsilon$ 是说，存在一个多项式时间的算法 A, 以一个实例 x 以及一个随机比特串 r 为输入，并输出 0 或 1。并且其需要满足:*

- 若 $x \in L$, 则 $\Pr(A(x,r)=1) = 1$
- 若 $x \notin L$, 则 $\Pr(A(x,r)=0) \geq 1-\epsilon$

定义 3 ($\mathcal{BPP}_\epsilon$)**.** *语言 $L \in \mathcal{BPP}_\epsilon$ 是说，存在一个多项式时间的算法 A, 以一个实例 x 以及一个随机比特串 r 为输入，并输出 0 或 1。并且其需要满足:*

- 若 $x \in L$, 则 $\Pr(A(x,r)=1) \geq 1-\epsilon$
- 若 $x \notin L$, 则 $\Pr(A(x,r)=0) \geq 1-\epsilon$

具体地，有定理 4 以及定理 5。

定理 4. *若常数 $\epsilon_1, \epsilon_2 \in (0,1)$, 则 $\mathcal{RP}_{\epsilon_1} = \mathcal{RP}_{\epsilon_2}$, co-$\mathcal{RP}_{\epsilon_1} =$ co-$\mathcal{RP}_{\epsilon_2}$。*

证明　不妨假设 $\epsilon_1 < \epsilon_2$. 只证 $\mathcal{RP}_{\epsilon_1} = \mathcal{RP}_{\epsilon_2}$, 对于 co-$\mathcal{RP}$ 来说，并无本质区别。

- 任何一个 $\mathcal{RP}_{\epsilon_1}$ 中的语言 L 都有一个单边错误率小于 ϵ_1 的多项式时间算法来决定它，这个算法错误率当然也小于 ϵ_2，因此 $L \in \mathcal{RP}_{\epsilon_2}$。从而 $\mathcal{RP}_{\epsilon_1} \subseteq \mathcal{RP}_{\epsilon_2}$
- 另一方面，假设 $L \in \mathcal{RP}_{\epsilon_2}$, 我们证明 $L \in \mathcal{RP}_{\epsilon_1}$. 由于 $L \in \mathcal{RP}_{\epsilon_2}$，存在一个多项式时间的算法 A, 使得当 $x \in L$ 时, $\Pr(A(x,r)=1) \geq 1-\epsilon_2$, $x \notin L$ 时，$\Pr(A(x,r)=0)=1$。以 A 作为子程序，构造算法 $\tilde{A}_k$(这里 k 是一个之后再确定的常数参数):$\tilde{A}_k$ 独立地选取 k 个随机比特串 $r_1, r_2, \ldots, r_k$，调用 k 次 A 计算 $A(x,r_1), A(x,r_2), ..., A(x,r_k)$. 如果存在 i 使得 $A(x,r_i)=1$，那么输出 1，否则输出 0。

显然 $\tilde{A}_k$ 的运行时间至多是 A 的 k 倍, 因此它也是多项式时间的。当 $x \notin L$ 时，每一个 $A(x, r_i)$ 都以概率 1 等于 0，因此 $\tilde{A}(x) = 0$。当 $x \in L$ 时

$$\begin{aligned}
\Pr(\tilde{A}_k(x) = 0) &= \Pr(\cap_{i=1}^{k} A(x, r_i) = 0) \\
&= \prod_{i=1}^{k} \Pr(A(x, r_i) = 0) && \text{独立性} \\
&\le \epsilon_2^k
\end{aligned}$$

取 $k \ge \frac{\ln(\epsilon_1)}{\ln(\epsilon_2)}$, 则有

$$\Pr(\tilde{A}_k(x) = 0) \le \epsilon_1$$

从而 $L \in \mathcal{RP}_{\epsilon_1}$，于是 $\mathcal{RP}_{\epsilon_2} \subseteq \mathcal{RP}_{\epsilon_1}$

□

Remark. 我们也可以把 $\tilde{A}$ 视作一个需要额外输入一个随机比特串的确定性算法，其随机比特串是子程序所需要使用的比特串的拼接，于是可以写作 $\tilde{A}_k(x, r_1, r_2, ..., r_k)$.

定理 5. 若 $\epsilon_1, \epsilon_2 \in (0, \frac{1}{2})$, 则 $\mathcal{BPP}_{\epsilon_1} = \mathcal{BPP}_{\epsilon_2}$。

证明 **不妨假设** $\epsilon_1 < \epsilon_2$。

- 一方面, $\mathcal{BPP}_{\epsilon_1} \subseteq \mathcal{BPP}_{\epsilon_2}$ 是显然的，因为错误率小于 ϵ_1 的算法自然也是错误率小于 ϵ_2 的算法。
- 另一方面，假设 $L \in \mathcal{BPP}_{\epsilon_2}$，我们证明 $L \in \mathcal{BPP}_{\epsilon_1}$。我们记 L 的一个错误率小于 ϵ_2 的判定算法为 A。以 A 作为子程序, 构造新的算法 $\tilde{A}_k$: $\tilde{A}_k$ 独立地选取 $2k+1$ 个随机比特串 $r_1, r_2, ..., r_{2k+1}$, 调用 $2k+1$ 次 A 计算 $A(x, r_1), A(x, r_2), ..., A(x, r_{2k+1})$。令 $\mathbb{X} = \sum_{i=1}^{2k+1} A(x, r_i)$，若 $\mathbb{X} \ge k+1$, 输出 1; 若 $\mathbb{X} \le k$, 输出 0.

  对任意 k，$\tilde{A}_k$ 的运行时间不过是 $2k+1$ 倍 A 的运行时间，因此它也是一个多项式时间算法。

  当 $x \in L$ 时

$$\begin{aligned}
\Pr(\tilde{A}_k(x) = 1) &= 1 - \Pr(\tilde{A}_k(x) = 0) \\
&= 1 - \Pr(\mathbb{X} \le k) \\
&= 1 - \Pr(\mathbb{X} - \mathbb{E}(\mathbb{X}) \le k - \mathbb{E}(\mathbb{X}))
\end{aligned}$$

  由于各个 $A(x, r_i)$ 是独立的，则

$$\mathbb{E}(\mathbb{X}) = \sum_{i=1}^{2k+1} \mathbb{E}(A(x, r_i)) \ge (2k+1)(1-\epsilon_2) = (2-2\epsilon_2)k + 1 - \epsilon_2$$

  我们还可以估计 $\mathbb{X}$ 的方差

$$\mathrm{Var}(\mathbb{X}) = \sum_{i=1}^{2k+1} \mathrm{Var}(A(x, r_i)) \le \sum_{i=1}^{2k+1} \frac{1}{4} = \frac{2k+1}{4}$$

注意到这里的不等式是因为，任何一个取值属于 $\{0,1\}$ 的随机变量，假设其取 1 得概率为 p，则其方差为 $p(1-p) \leq \frac{1}{4}$。

于是

$$
\begin{aligned}
\Pr(\tilde{A}_k(x)=1) &= 1-\Pr(\mathbb{X}-\mathbb{E}(\mathbb{X}) \leq k-\mathbb{E}(\mathbb{X})) \\
&\geq 1-\Pr(\mathbb{X}-\mathbb{E}(\mathbb{X}) \leq (2\epsilon_2-1)k+\epsilon_2-1) \\
&\geq 1-\Pr(\mathbb{X}-\mathbb{E}(\mathbb{X}) \leq (2\epsilon_2-1)k) \\
&\geq 1-\Pr(|\mathbb{X}-\mathbb{E}(\mathbb{X})| \geq (1-2\epsilon_2)k) && 2\epsilon_2-1<0 \\
&\geq 1-\frac{\mathrm{Var}(\mathbb{X})}{(1-2\epsilon_2)^2k^2} && \text{Chebyshev 不等式} \\
&\geq 1-\frac{2k+1}{4(1-2\epsilon_2)^2k^2}
\end{aligned}
$$

注意到 $1-2\epsilon>0$，所以 $\frac{2k+1}{4(1-2\epsilon_2)^2k^2}=\mathrm{O}\left(\frac{1}{k}\right)$ 随 k 增大从正数趋向于 0，可知一定存在足够大的 k_1, 使得 $\Pr(\tilde{A}_{k_1}(x)=1) \geq 1-\epsilon_1$

当 $x \notin L$ 时，分析是完全一样的，可以知道

$$\Pr(\tilde{A}_k(x)=0) \geq 1-\frac{2k+1}{4(1-2\epsilon_2)^2k^2}$$

取上述的 k_1 时，有 $\Pr(\tilde{A}_{k_1}(x)=0) \geq 1-\epsilon_1$。于是 $L \in \mathcal{BPP}_{\epsilon_1}$, 从而 $\mathcal{BPP}_{\epsilon_2} \subseteq \mathcal{BPP}_{\epsilon_1}$

综上我们证明了 $\mathcal{BPP}_{\epsilon_1}=\mathcal{BPP}_{\epsilon_2}$。 □

1.2 Las Vegas 算法和 Monte Carlo 算法

Las Vegas 算法 一个 Las Vegas 算法总能输出正确的结果，它的运行时间是随机的，可能会非常大，但是这个运行时间的期望是有限的。

Monte Carlo 算法 一个 Monte Carlo 算法的运行时间是有严格的上界的，但是它不一定总能输出正确的结果。

Las Vegas 算法和 Monte Carlo 算法总是可以相互转换的，换言之，如果我们有一个问题的 Las Vegas 算法，那么我们就可以得到这个问题的一个 Monte Carlo 算法，反之亦然。

1.3 $\mathcal{ZPP}$

通过前面的学习, 我们知道 $\mathcal{RP}$, co-$\mathcal{RP}$ 和 $\mathcal{BPP}$ 对应着问题有 Monte Carlo 算法。然而 Las Vegas 算法的复杂性类对应物是什么呢，这就是接下来介绍的 $\mathcal{ZPP}$。

定义 6. *称语言 $L \in \mathcal{ZPP}$，当且仅当对 L 存在期望运行时间为多项式时间的随机算法 A，A 以 x 为输入，并使用若干位随机比特串 r，使得*

$$x \in L \iff A(x,r)=1.$$

我们提到 Las Vegas 算法和 Monte Carlo 算法可以相互转换，这件事情严谨地说就是下面这个定理。注意到由于我们已经证明了 $\mathcal{RP}$ 和 co-$\mathcal{RP}$ 定义中的错误率并不重要，下面定理中的 $\mathcal{RP}$ 和 co-$\mathcal{RP}$ 的错误率就简单设为 $\frac{1}{2}$。

定理 7. $\mathcal{ZPP} = \mathcal{RP} \cap co\text{-}\mathcal{RP}$

证明 首先证明 $\mathcal{ZPP} \subseteq \mathcal{RP} \cap \text{co-}\mathcal{RP}$。下证 $\mathcal{ZPP} \subseteq \mathcal{RP}$，即证 $\forall L \in \mathcal{ZPP} \Longrightarrow L \in \mathcal{RP}$。

首先由 $\mathcal{ZPP}$ 定义，$L \in \mathcal{ZPP}$ 意味着存在一个期望运行时间是多项式时间的算法 A，使得

$$x \in L \iff A(x, r) = 1.$$

设算法 A 的期望运行时间为 $T(n)$，n 是实例的规模，现在构造随机算法 $\tilde{A}$ 如下：$\tilde{A}$ 接收 x 作为输入，并使用随机比特串 r，只调用算法 $A(x, r)$ 一次，若 A 在 $10T(n)$ 步之内停机，则算法输出 $A(x, r)$；否则，算法输出 0。

显然，算法 $\tilde{A}$ 一定会在 $10T(n)$ 时间内终止，因此他是一个多项式时间算法。且易知 $x \notin L \Longrightarrow \Pr(\tilde{A}(x, r) = 0) = 1$。下证 $x \in L \Longrightarrow \Pr(\tilde{A}(x, r) = 1) \geq \frac{1}{2}$。

设算法 $\tilde{A}$ 某次调用 A 时实际运行时间为 T，则当 $x \in L$ 时，

$$\begin{aligned}
\Pr(\tilde{A}(x, r) = 1) &= 1 - \Pr(\tilde{A}(x, r) = 0) \\
&= 1 - \Pr(T > 10T(n)) \\
&\geq 1 - \frac{T(n)}{10T(n)} && \text{Markov 不等式} \\
&\geq \frac{9}{10} > \frac{1}{2}.
\end{aligned}$$

于是可知 $L \in \mathcal{RP}$。对于 co-$\mathcal{RP}$, 只需要让 $\tilde{A}$ 在调用 A $10T(n)$ 步后仍然没有得出结果时输出 1 而不是 0，就可以同理证明 $L \in \text{co-}\mathcal{RP}$。于是 $\mathcal{ZPP} \subseteq \mathcal{RP} \cap \text{co-}\mathcal{RP}$。

下证 $\mathcal{RP} \cap \text{co-}\mathcal{RP} \subseteq \mathcal{ZPP}$, 即证 $\forall L \in \mathcal{RP} \cap \text{co-}\mathcal{RP} \Longrightarrow L \in \mathcal{ZPP}$。

根据类 $\mathcal{RP}$ 和 co-$\mathcal{RP}$ 的定义，$L \in \mathcal{RP} \cap \text{co-}\mathcal{RP}$ 意味着存在一个多项式时间随机算法 A_1，A_1 以 x 为输入，并使用若干位随机比特 r，使得

$$\begin{aligned}
x \in L &\Longrightarrow \Pr(A_1(x, r) = 1) \geq \frac{1}{2}, \\
x \notin L &\Longrightarrow \Pr(A_1(x, r) = 0) = 1.
\end{aligned}$$

同时存在一个多项式时间随机算法 A_2，A_2 以 x 为输入，并使用若干位随机比特 r'，使得

$$\begin{aligned}
x \in L &\Longrightarrow \Pr(A_2(x, r') = 1) = 1, \\
x \notin L &\Longrightarrow \Pr(A_2(x, r') = 0) \geq \frac{1}{2}.
\end{aligned}$$

我们想要构造一个以概率 1 输出正确结果的算法 $\tilde{A}$，它具有期望多项式运行时间。可以观察到，虽然算法 A_1, A_2 都不保证输出正确结果，但它们都只有单边错误。只要 A_1 输出 1，就一定有 $x \in L$；同样地，只要 A_2 输出 0，就一定有 $x \notin L$。通过这个规律，可以构造随机算法 $\tilde{A}$ 如下：

$\tilde{A}$ 接收 x 作为输入,它的第 i 轮独立地产生两个随机比特串 r_i, r'_i、并且分别调用算法 $A_1(x, r_i), A_2(x, r'_i)$，若 $A_1(x, r_i)$ 输出 1，则算法输出 1；若 $A_2(x, r'_i)$ 输出 0，则算法输出 0；否则，算法进入第 $i+1$ 轮。

算法 $\tilde{A}$ 中每轮的分支都不会产生冲突，因为 $A_1(x, r_i) = 1$ 和 $A_2(x, r'_i) = 0$ 不可能同时发生。另外，根据上面的分析，$\tilde{A}$ 只要停机，就会以概率 1 输出正确结果。接下来我们需要考虑 $\tilde{A}$ 的期望运行时间。

记事件 H_i 为第 i 轮两个子程序输出的结果能够使 $\tilde{A}$ 停机，则 $\Pr(H_i) = \Pr(A_1(x, r_i) = 1 \text{ or } A_2(x, r_i') = 0)$。当 $x \in L$ 时，

$$\begin{aligned}&\Pr(A_1(x, r_i) = 1 \text{ or } A_2(x, r_i') = 0)\\ \geq &\Pr(A_1(x, r_i) = 1)\\ \geq &\frac{1}{2}.\end{aligned}$$

同样地，当 $x \notin L$ 时，

$$\begin{aligned}&\Pr(A_1(x, r_i) = 1 \text{ or } A_2(x, r_i') = 0)\\ \geq &\Pr(A_2(x, r_i') = 0)\\ \geq &\frac{1}{2}.\end{aligned}$$

设 $\tilde{A}$ 的运行时间为 T，A_1, A_2 的运行时间分别为 T_1, T_2，利用几何分布的期望，有 $\mathbb{E}(T) \leq 2O(T_1+T_2)$，所以 $\tilde{A}$ 具有多项式的期望运行时间。从而 $L \in \mathcal{ZPP}$，于是 $\mathcal{ZPP} \subseteq \mathcal{RP} \cap \text{co-}\mathcal{RP}$。 □

2 球盒模型 (Balls & Bins Model)

在第一堂课上，我们学习到了生日悖论，在这里我们把生日悖论这个问题模型加以推广，这就是球盒模型。假设我们有 m 个球，有 n 个盒子，对于每一个球，我们都独立地，均匀随机地扔到某一个盒子中，也就是说扔到每个盒子的概率都是 $\frac{1}{n}$。记 X_i 为盒子 i 中的小球数量，称为盒子 i 的负载 (load)。球盒问题就是研究这些随机变量的表现。注意到当 $n = 365$, m 为教室里的学生人数时，这就是生日悖论在讨论的事，也就是求事件 $\max_i X_i \geq 2$ 的概率。

Balls & Bins 模型在很多随机算法设计问题中都有应用，例如服务器负载均衡，哈希碰撞等等。

2.1 最大负载

在这里，我们考虑 $m = n$，即球和箱子的数量相等。我们想知道此时含有最多球的箱子中大概有多少球。换言之，令 $X = \max\limits_{1 \leq i \leq n} X_i$，我们想知道 $\mathbb{E}(X)$ 的量级。实际上，我们会看到 $\mathbb{E}(X) = \Theta(\frac{\ln n}{\ln \ln n})$，我们将证明的结论比这个更强:$X = \Theta(\frac{\ln n}{\ln \ln n})$ 以高概率成立。

Remark. $f = \Theta(g)$ 意味着 $f = \Omega(g)$ 且 $f = \mathrm{O}(g)$。即 f 与 g 基本只差一个常数。

下面我们证明这一点。首先我们给出两个引理。

引理 8 (Union Bound)**.** 对于有限个事件 $A_1, A_2, ..., A_n$

$$\Pr\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} \Pr(A_i)$$

证明 我们只需对两个事件证明即可，剩下的可以用数学归纳法证得。由容斥定理

$$\Pr(A_1 \cup A_2) = \Pr(A_1) + \Pr(A_2) - \Pr(A_1 \cap A_2) \leq \Pr(A_1) + \Pr(A_2)$$

得证。 □

引理 9. 对于组合数 $\binom{n}{k}$，我们有如下不等式

$$\left(\frac{n}{k}\right)^k \le \binom{n}{k} \le \left(\frac{en}{k}\right)^k$$

作业. 证明引理 9.

定理 10. *(最大负载)* 当球数和箱子数相等，都为 n 时，以高概率有 $X = \Theta(\frac{\ln n}{\ln \ln n})$，即

$$\Pr\left(X = \mathrm{O}\left(\frac{\ln n}{\ln \ln n}\right)\right) = 1 - o(1)$$
$$\Pr\left(X = \Omega\left(\frac{\ln n}{\ln \ln n}\right)\right) = 1 - o(1)$$

证明 下面证明

$$\Pr\left(X < 4\frac{\ln n}{\ln \ln n}\right) = 1 - o(1)$$
$$\Pr\left(X > \frac{1}{4}\frac{\ln n}{\ln \ln n}\right) = 1 - o(1)$$

不妨记 $\frac{\ln n}{\ln \ln n} = S$

- **上界.** 由于 $\Pr(X < 4S) = 1 - \Pr(X \ge 4S)$。我们先计算 $\Pr(X \ge 4S)$.

  先考虑 $X_1 \ge 4S$ 的概率。我们可以将盒子中有不少于 $4S$ 个球拆分成一些事件的并，即给出具体的 $4S$ 个球扔到了盒子中，注意这些事件不一定是互斥的，但这不影响我们接下来的论证。

$$\begin{aligned}
\Pr(X_1 \ge 4S) &= \Pr\left(\bigcup_{1\le j_1<\ldots<j_{4S}\le n} \#j_1,\ldots,j_{4S} \text{ balls } \to \#1 \text{ bin}\right) \\
&\le \sum_{1\le j_1<\ldots<j_{4S}\le n} \Pr\left(\#j_1,\ldots,j_{4S} \text{ balls} \to \#1 \text{ bin}\right) && \text{Union Bound} \\
&= \binom{n}{4S}\cdot\left(\frac{1}{n}\right)^{4S} \\
&\le \left(\frac{ne}{4S}\right)^{4S}\cdot\left(\frac{1}{n}\right)^{4S} && \text{引理 9} \\
&< \left(\frac{1}{S}\right)^{4S} = \left(\frac{\ln \ln n}{\ln n}\right)^{4S} && \text{约去}n^{4S}\text{并且}\frac{e}{4}<1 \\
&< \left(\frac{\sqrt{\ln n}}{\ln n}\right)^{4S} = \left(\frac{1}{\ln n}\right)^{2S} \\
&= (\ln n)^{-2\frac{\ln n}{\ln \ln n}} = \left(e^{\ln \ln n}\right)^{-2\frac{\ln n}{\ln \ln n}} = \frac{1}{n^2}
\end{aligned}$$

  注意到对于每个盒子，上述不等式都成立。那么

$$\begin{aligned}
\Pr(X \ge 4S) &= \Pr((X_1 \ge 4S) \cup (X_2 \ge 4S) \cup \ldots \cup (X_n \ge 4S)) \\
&\le \sum_{i=1}^{n} \Pr(X_i \ge 4S) < n\cdot\frac{1}{n^2} = \frac{1}{n} && \text{Union Bound}
\end{aligned}$$

  因此 $\Pr\left(X < 4\left(\frac{\ln n}{\ln \ln n}\right)\right) = 1 - \Pr(X \ge 4S) > 1 - \frac{1}{n} = 1 - o(1)$.

- **下界.** 我们先考虑某一个盒子 (例如第一个) 中球的数量，我们要证明这个盒子中球的数量大于 $\frac{1}{4}S$ 的概率不会太低。即

$$\begin{aligned}
\Pr\left(X_1 > \frac{1}{4}S\right) &\geq \Pr\left(X_1 \geq \frac{1}{3}S\right) \geq \Pr\left(X_1 = \frac{1}{3}S\right) \\
&= \binom{n}{\frac{S}{3}}\left(\frac{1}{n}\right)^{\frac{S}{3}}\left(1-\frac{1}{n}\right)^{n-\frac{S}{3}} \\
&\geq \binom{n}{\frac{S}{3}}\left(\frac{1}{n}\right)^{\frac{S}{3}}\left(1-\frac{1}{n}\right)^{n} \\
&\geq \left(\frac{n}{\frac{S}{3}}\right)^{\frac{S}{3}}\left(\frac{1}{n}\right)^{\frac{S}{3}}\frac{1}{4} = \left(\frac{3}{S}\right)^{\frac{S}{3}}\frac{1}{4} && \text{引理 9} \\
&= \frac{1}{4}\cdot\left(\frac{3\ln\ln n}{\ln n}\right)^{\frac{\ln n}{3\ln\ln n}} \\
&\geq \frac{1}{4}\left(\frac{1}{\ln n}\right)^{\frac{\ln n}{3\ln\ln n}} = \frac{1}{4}\cdot\left(e^{-\ln\ln n}\right)^{\frac{\ln n}{3\ln\ln n}} \\
&= \frac{1}{4}n^{-1/3}
\end{aligned}$$

在上面的公式推导中，我们还利用了 $\left(1-\frac{1}{n}\right)^n$ 的单调性，将其放缩到 $\left(1-\frac{1}{2}\right)^2 = \frac{1}{4}$。引入 0-1 随机变量 Y_i

$$Y_i = \begin{cases} 1 & \text{if } X_i > \frac{S}{4} \\ 0 & \text{otherwise} \end{cases}$$

令

$$Z = Y_1 + Y_2 + \cdots + Y_n$$

我们先计算 Z 的期望与方差

$$\mathbb{E}(Z) = \sum_{i=1}^{n}\mathbb{E}(Y_i) \geq n\cdot\frac{1}{4}n^{-1/3} = \frac{1}{4}n^{2/3}$$

对于方差，我们需要注意到各个 Y_i 并不是独立的，因此

$$\mathrm{Var}(Z) = \sum_{i=1}^{n}\mathrm{Var}(Y_i) + \sum_{i\neq j}\mathrm{Cov}(Y_i, Y_j)$$

这里的 Cov 是协方差。由约束 $\sum_{i=1}^{n} X_i = n$ 可以知道，若某个 X_i 更大，那么其它的 X_i 会更小，则 Y_i 间是负相关的，从而 $\mathrm{Cov}(Y_i, Y_j) \leq 0\ \forall i \neq j$，又由于 $0, 1$ 随机变量的方差不超过 $\frac{1}{4}$，于是

$$\mathrm{Var}(Z) \leq \sum_{i=1}^{n}\mathrm{Var}(Y_i) \leq \frac{n}{4}$$

现在，有了 Z 的期望与方差的估计，我们可以应用 Chebyshev 不等式来计算 $\Pr\left(X > \frac{S}{4}\right)$。

$$\begin{aligned}
\Pr\left(X > \frac{S}{4}\right) &= \Pr\left(\bigcup_{i=1}^{n}\left(X_i > \frac{S}{4}\right)\right) \\
&= \Pr(Z > 0) = 1 - \Pr(Z = 0)
\end{aligned}$$

而

$$\begin{aligned}
\Pr(Z=0) &= \Pr(Z-\mathbb{E}(Z)=-\mathbb{E}(Z)) \\
&\le \Pr(|Z-\mathbb{E}(Z)| \ge |\mathbb{E}(Z)|) \\
&\le \frac{\mathrm{Var}(Z)}{|\mathbb{E}(Z)|^2} \\
&\le \frac{\frac{1}{4}n}{\left(\frac{1}{4}n^{\frac{2}{3}}\right)^2} \\
&= 4n^{-1/3} = o(1)
\end{aligned}$$

因此

$$\Pr\left(X > \frac{S}{4}\right) = 1 - o(1)$$

综上所述，我们最终证明了 $X = \Theta(\frac{\ln n}{\ln\ln n})$ 以高概率成立。 □