计算理论大作业——读书报告

姓名：钱泽凯

学号：1190202011

目录

Pseudorandomness 读书报告

伪随机理论指的是尽管构造很少或没有使用随机性，但是还是有效生成 "看起来是随机的 "对象的理论。这一理论对计算机科学和数学中的许多领域都有意义，包括计算复杂性、算法、密码学、组合学、信息学和数论。

Salil Vadhan 所著经典 pseudorandomness 在相关伪随机领域相当重要。文章的的关注点特别强调了在各种基本的“伪随机对象”之间发现的密切联系，同时这些基本的“伪随机对象”在扩展图、随机抽取器、列表可解错误校正码、取样器和伪随机发生器的性质上似乎有很大的不同。

下面就前三章内容，结合本学期（有些是大二下学习的高级算法）所学内容，写一篇读书报告。

一、 Chapter 1: Introduction

1. 普遍应用

随机性是过去计算机科学中最普遍的范例，有极多的应用。

(i) 算法设计

Primality Testing, Approximate Counting, Undirected S-T Connectivity、完美匹配等问题中，已知最有效的问题的算法是随机化的。即使后来找到了复杂度在一个数量级的算法，随机算法也更简单和更容易理解。

(ii) 密码学

随机性保障密码的秘密性，因此随机化是密码学的核心，包括密钥和加密算法都必须是随机化的，这才能保证足够的安全性。

(iii) 组合构造

凭借证明一个对象具有概率非 0 的某个属性，表明一些组合结构的存在性，如 Ramsey Graph 的存在性：一个随机选择 n 个顶点的随机生成图，没有大小为 2logn 的团或者独立集的概率很高。通过这个性质，将在后续章节扩展概率方法的其他的两个应用：扩展图和纠错码。

2. 消除随机化的影响

现在的问题是去随机化后，能够做到随机性做到的东西么？有以下几个方面值得考虑：

(i) 复杂性理论

加入允许算法有较小的效率下降（包括时间、空间、并行性），是否所有随机算法都存在去随机化版本？

(ii) Using Physical Random Sources

由于目前还无法知道现实世界是否有物理物质可以表现纯随机性，所以基于物理的随机数生成器获得的随机数将有一定的偏差，更多的表现为彼此相关，如何处理生成的这些数？

(iii) 显式构造

随机性能够证明存在性，但是通常随机选择的对象具有指数级别的参数描述，那么这样的存在性能否用构造的实例出来呢？

(iv) 意料之外的应用

与去随机化无关的领域里，也有出现涉及伪随机性的应用，包括数据结构，分布式计算，复杂性理论等，将在后续章节中逐步介绍。

3. 研究目标和课题

(i) 伪随机数生成器

其基于少量真正随机的 seed 生成大量的随机数，生成的数据不可能是完全随机的。但是没有一个有效的算法可以将这种伪随机数序列的输出与真正的随机序列区分开来，需要进行条件性假设，理想的定理是：“如果 P≠NP，则伪随机数生成器存在”。在去随机化算法中，这些至关重要。

(ii) Randomness Extractors

随机数提取器就主要关注第 2 点提到的物理源的相关性问题了，内容是把相关的、有偏差的输入转换成几乎均匀分布的伪随机数。

(iii) Expander Graphs

扩展图是一种强连通的稀疏图，这是一种看似“矛盾”的概念，它的实际意义是一个模型“连接良好”：如果不把很大一部分边切断，该图就不会被完全一分为二。证明其存在性并不明显，但是事实上可以采用第 1 点应用中提到的组合构造来进行证明，即采用概率的方法，证明一个 3 度的随即图是一个高概率存在的扩展图。但是我们更希望一个显式的去随机化的构造方案。后续章节中，我们将看到一些显式的构造手段，但是即使目前最先进的构造也不总是与随机化方法给出的概率相当。

扩展图的应用初始被设计用于容错网络中（如电话线），即在一些节点失效后仍能保证一定的连接性。同时在一些并不明显的应用中也有应用，如 O(logn)的并行排序算法。

(iv) Error-Correcting Codes

纠错码是一种在有噪声信道上通信的工具。其将一串信息编码为更长的

信息，保证这些更长的信息在受到干扰的时候，仍然能复原正确的解码。目前香农用随机化的概率方法证明了“好”纠错码的存在，即将 n 位比特信息随机映射到 O(n)位码是一种高概率的“好”纠错码。但是仍然无法获取这样的纠错码，因为随机映射是一种指数大小的对象，我们无法合理的解码这样的映射。

我们仍然需要显式构造，既然随机映射的解码列表是指数级别大小，那我们研究的重点则是：如何生成一个尽量简短的解码列表，能包括和还原正确的信息。

4. 总结

以上 4 个课题尽管过去分属于不同的领域，但是在一定解释下，本质是相同的课题，伪随机数发生器通过断言高效的算法的缺陷可以作为复杂性理论的对象；随机数提取器重点是关注序列中有偏差的熵，可以作为信息论的对象；拓展图问题当然输入组合构造对象（见第 1 点所述）；而纠错码问题则是组合学、信息论、复杂性代数的混合问题。后面将重点关注他们的联系，从而对每一个问题有新的理解和进展。

二、 Chapter 2: The Power of Randomness

本章将主要介绍一些去随机化的算法的例子，以及研究复杂性理论和随机化算法的基本工具。主要通过一些随机化方法介绍了两个工具，其一是尾部定界，即马尔可夫不等式(Markov Inequality)和切尔诺夫界(Chernoff Bound)、其二是用代数的方法研究图论，即随机游走。

1. Polynomial Identity Testing

(i) 多项式相等问题

首先是多项式同一性测试这一问题。定义如下：

Computational Problem 2.1. POLYNOMIAL IDENTITY TESTING: Given two multivariate polynomials, $f(x_1,\ldots,x_n)$ and $h(x_1,\ldots,x_n)$, decide whether $f = g$.

图 2.1 多项式同一性定义

注意的是这里的 f=g 只得是各个系数一致，即形式相等，并不是值相等。这两者在有限域是等价的，但是在诸如 $f(x) = \prod_{\alpha\in\mathbb{F}}(x-\alpha).$ 的无限域下不等价，因为后者可能不可计算。我们用等价的版本来进行后续的工作：

Computational Problem 2.2. POLYNOMIAL IDENTITY TESTING (reformulation): Given a polynomial $f(x_1,\ldots,x_n)$, is $f = 0$?

图 2.2 多项式同一性等价命题

为了解决这个问题需要使用 Schwartz 定理，如下所示，描述了 f=0 的概率情况。

Lemma 2.4 (Schwartz–Zippel Lemma). If f is a nonzero polynomial of degree d over a field (or integral domain) $\mathbb{F}$ and $S \subset \mathbb{F}$, then

$$\Pr_{\alpha_1,\ldots,\alpha_n \overset{R}{\leftarrow} S}[f(\alpha_1,\ldots,\alpha_n) = 0] \leq \frac{d}{|S|}.$$

图 2.3 Schwartz 定理

因此采用以下随机化算法即可，该算法的错误概率为 1/2，多次重复算法或者增加 S 的范围就能降低错误的概率。同时能应用这个算法有两个条件，一个是可以估计或者计算出多项式的度数 d，另一个是可以在多项式时间内计算 f 的值。

Algorithm 2.3 (Polynomial Identity Testing).
Input: A multivariate polynomial $f(x_1,\ldots,x_n)$ of degree at most d over a field/domain $\mathbb{F}$.

(1) Let $S \subset \mathbb{F}$ be any set of size $2d$.
(2) Choose $\alpha_1,\ldots,\alpha_n \overset{R}{\leftarrow} S$.
(3) Evaluate $f(\alpha_1,\ldots,\alpha_n)$. If the result is 0, accept. Otherwise, reject.

图 2.4 多项式同一性随机算法

(ii) 完美匹配问题

在解决了多项式的同一性问题后，可以根据词来解决图的完美匹配的问题。图的匹配是指，对于图的一个边的子集，这个集合中每条边的两端点没有公共点，则这些边和点构成了一个匹配。图的完美匹配是指若图中的一个匹配，包括了图中的所有点，则称这个匹配为完美匹配。完美匹配使图中所有点都为匹配点。

如下算法可以一个 Tutte 矩阵。Tutte 定理指出，图中有完美匹配等价于 Tutte 矩阵的行列式不为 0。

Algorithm 2.7 (PERFECT MATCHING **in bipartite graphs).**
Input: A bipartite graph G with vertices numbered $1,\ldots,n$ on each side and edges $E \subseteq [n] \times [n]$.[2]

We construct an $n \times n$ matrix A where

$$A_{i,j}(x) = \begin{cases} x_{i,j} & \text{if}(i,j) \in E \\ 0 & \text{otherwise} \end{cases},$$

where $x_{i,j}$ is a formal variable.

Consider the multivariate polynomial

$$\det(A(x)) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \cdot \prod_i A_{i,\sigma(i)},$$

图 2.5 完美匹配问题算法

2. The Computational Model and Complexity Classes

为了严格的定义随机算法理论，需要一个精准的计算模型定义，下面复习几个复杂性类。

(i) RP

RP(Randomize Polynominal Time)，存在单边错误率, 即若 L 属于 BPP，则存在一个算法 A，存在 c，在多项式时间 O(n^c)时间内可判定，有如下概率条件：

- $x \in L \Rightarrow \Pr[A(x) \text{ accepts}] \geq 1/2.$
- $x \notin L \Rightarrow \Pr[A(x) \text{ accepts}] = 0.$

co-RP 与 RP 问题不同的地方在于允许的错误率发生在 x 不属于 L。

(ii) BPP

BPP(Bounded error Probabilistic Polynominal Time)，存在双边错误率；即若 L 属于 BPP，则存在一个算法 A，存在 c，在多项式时间 O(n^c)时间内可判定。

Definition 2.13. BPP is the class of languages L for which there exists a probabilistic polynomial-time algorithm A such that

- $x \in L \Rightarrow \Pr[A(x) \text{ accepts}] \geq 2/3.$
- $x \notin L \Rightarrow \Pr[A(x) \text{ accepts}] \leq 1/3.$

图 2.6 BPP 定义

(iii) ZPP

前面的类都是在多项式时间内，算法运行必须停止，且得到一个输出，只是有的输出存在一定的错误率；ZPP 的多项式时间是一个期望时间，即算法可能无限的运行下去，但得到的结果是没有错误率的。同时有如下定理：

Fact 2.16 (Problem 2.3). ZPP = RP ∩ co-RP.

图 2.7 ZPP 定理

3. Tail Inequalities and Error Reduction

除了《伪随机性》这本书本章的内容外，我们在大二下半学期学习的高级算法课程中还介绍了诸如切比雪夫不等式等概率定界方法，下面一起进行总结。

可以使用尾部不等式来约束概率分布尾部的值应被视为异常的概率。尾部不等式的强度取 决于对基础随机变量的假设数量。 较少的假设会导致较弱的不等式，但这种不等式适用于 较大类别的随机变量。例如，Markov 和 Chebychev 不等式是弱不等式，但它们适用于非常 大类的随机变量。另一方面，Chernoff bound 和 Hoeffding 不等式都是更强的不等式，但它们适用于受限类别的随机变量。

马尔可夫不等式是最基本的尾部不等式之一，它定义为仅采用非负值的分布。令 X 为随机变量，概率分布为 f（X），均值为 E[X]，方差为 Var[X]。马尔可夫不等式仅针对非负值的概率分布而定义，并且仅在上尾提供界限。在实践中，通常希望绑定任意分布的两个尾部。考虑 X 是任意随机变量的情况，其不一定是非负的。在这种情况下，不能直接使用马尔可夫不等式。

Lemma 2.20 (Markov's Inequality). If X is a nonnegative real-valued random variable, then for any $\alpha > 0$,

$$\Pr[X \geq \alpha] \leq \frac{\mathrm{E}[X]}{\alpha}.$$

图 2.8 马尔可夫不等式

然而，（相关的）切比丘夫不等式在这种情况下非常有用。Chebychev 不等式是马尔可夫不等式直接应用于随机变量 X 的非负导数，如下图所示。

Lemma 3.27 (Chebyshev's Inequality). If X is a random variable with expectation μ, then

$$\Pr[|X - \mu| \geq \varepsilon] \leq \frac{\mathrm{Var}[X]}{\varepsilon^2}$$

图 2.9 切比雪夫不等式

马尔可夫和切比切夫不等式是相对较弱的不等式， 并且通常不能提供足够严格的界限以在许多实际情况中有用。 这是因为这些不等式不对随机变量 X 的性质做出任何假设。 然而，当对随机变量使用更强的假设时，可以捕获许多 实际情况。 在这种情况下，尾部分布的边界可能更紧密。特定情况是随机变量 X 可以表示为其他独立有界随机变量之和的情况。还有 Chernoff 界进行定界。

Theorem 2.21 (A Chernoff Bound). Let $X_1, \ldots, X_t$ be independent random variables taking values in the interval $[0,1]$, let $X = (\sum_i X_i)/t$, and $\mu = \mathrm{E}[X]$. Then

$$\Pr[|X - \mu| \geq \varepsilon] \leq 2\exp(-t\varepsilon^2/4).$$

图 2.10 切尔诺夫界

可以将切尔诺夫界应用到 BPP 上，有如下定理，即一个语言输入 BPP 等价于对于任意一个多项式 p，都有一个概率多项式时间算法，双面误差最多为 2^p；或者存在有一个多项式 q，有多项式时间算法的双面误差为 1/2-1/q;

Proposition 2.22. The following are equivalent:

(1) $L \in \mathbf{BPP}$.
(2) For every polynomial p, L has a probabilistic polynomial-time algorithm with two-sided error at most $2^{-p(n)}$.
(3) There exists a polynomial q such that L has a probabilistic polynomial-time algorithm with two-sided error at most $1/2 - 1/q(n)$.

图 2.11 BPP 等价定理

4. 随机游走

(i) 随机游走的性质

随机游走（random walk）也称随机漫步，随机行走等是指基于过去的表现，无法预测将来的发展步骤和方向。随机游走即从某个点出发, 然后在该点的边(如果 digraph 则为出边)中均匀随机选择一条, 完成一步移动. 并重复此过程. 这个过程虽然简单, 但是对它进行分析可以解决很大一类问题, 因此具有研究的价值。

伪随机性研究的一个重要工具就是谱图论. 它允许我们用代数的手段来研究图论. 谱图论中, 我们经常用到 digraph 和无向图。digraph 即 directed multigrap

h, 在有向图的基础上, 允许多重边的存在. 且允许点到自身的边(称为自回路, self-loop)。多重无向图即在无向图的基础上, 允许多重边的存在, 且允自回路。

值得注意的是, 在随机游走的研究中, 我们认为点到自身的边只它贡献 1 度(degree). 这是由于我们需要用到 d-正则(regular)图的概念。d-正则图即图中每条边的度都是 d。我们经常需要将任意图补上若干自回路, 使其变成一个-正则图. 如果按照一般的图论, 认为自回路为点贡献 2 度, 那么这样的操作就无法完成。

值得注意的是随机游走矩阵 M 是个非常特殊的矩阵, 具有一些非平凡的性质:对于无向图来说, M 是个实对称矩阵, 这意味着它可以被对角化, 且可以找到 rank(M)个正交的特征向量.。显然, 不同特征值的特征向量是正交的;对于正则图来说, M 的各列之和为 1, 那么 uM=u 因此 u 是 M 的一个对应特征值 1 的特征向量;

对于无向正则图, 应该具有如下特点：首先 M 的特征值的绝对值最大为 1；其次 M 有关特征值 1 的特征空间为 1⇔G 是连通图；最后 M 有特征值−1⇔G 是二部图。

对于 d-正则有向图 G, 以及其随机游走矩阵, 应该具有如下特点：首先 u 是 M 的一个对应特征值 1 的特征向量；其次，G 的每一个弱连通分支都是强连通的. 这是由于如果一个弱连通分支, 这个现象很有趣, 因此如果 G 的某一个弱连通分支, 按照 DAG 的方式, 将强连通分支排出来。

(ii) 无向图的连通性问题

到此, 我们可以解决一个非常重要的问题, 即无向图点对连通性问题：给定无向图 G 和其中的两个点 s,t, 判定是否存在一条从 s 到 t 的通路。

显然, 深度优先搜索和宽度优先搜索都可以在多项式时间内解决这一问题, 而我们要考虑的是, 能否在对数空间内解决这一问题. 我们解决该问题用到的算法时随机的; 在连通的正规图中, 我们可以以多项式步的随机游走, 以极大的概率来触及到图中的每个点, 以这一思想设计出来的算法非常简单: 在图 G 中从 s 开始随机游走 poly(n)步, 当达到 t 时, 接受; 否则, 拒绝.为了更好的描述这个问题, 我们定义一个图 G 的 hitting time。

Definition 2.48. For a digraph $G=(V,E)$, we define its *hitting time* as

$$\mathrm{hit}(G) = \max_{i,j\in V} \min\{t : \Pr[\text{a random walk of length } t \text{ started at } i \text{ visits } j] \geq 1/2\}.$$

图 2.12 hitting time

简言之, 从任意点 i 开始, 经历 t 步随机游走后, 图上的任意点 j 被访问过的概率至少是 1/2. 显然, 对于图 G, 如果 hit(G)=poly(n), 那么进行多项式步随机

游走, 就可以压倒性地概率遍及每一个点. 那么, 我们如果证明了任何一个连通图 G 的 hit(G)是多项式, 就可以证明我们给出的随机游走的无向图通路算法是正确的. 有如下定理：

给定任意连通的无向图 G, 包含 n 个点, 其中点的最大的度为 d, 则有 hit(G)=O(d^2n^3logn).

任何一个给定的无向图, 我们都可以通过向其添加自回路, 使得该图变为一个 d-正则或(d+1)-正则的非二部图. 而这一操作只可能增大图的 hitting time(因为增大了每一步随机游走中, 停留在出发点的概率), 因此我们对正则非二部图证明上述结论就可以了。

三、 Chapter 3: Basic Derandomization Techniques

1. Enumeration（枚举）

对于 L∈BPP 的一个 PPT 算法 A，A 的随机比特数必然是 x 的多项式。可以用一个确定性算法来判定 L，首先枚举这 2^ploy(x)个随机比特，计算 A(x, r)。随后，统计满足 A(x,r)=1 所占的比例，如果达到 A 输出 1 的阈值要求则输出 1，否则输入 0。

由上面的证明，从而有如下的定理：

Proposition 3.2. BPP ⊂ EXP.

图 3.1 BPP 定理

同时，如果总是能缩减随机算法的随机 bit 到 O(logn)个，那么存在确定性算法，能够把随机算法在几乎相同的性能下去随机化。通俗来说，去随机化算法的最坏时间复杂度为去随机化之前的多项式倍。

2. Nonconstructive/Nonuniform Derandomization（非一致性方法）

我们可以证明 BPP∈P/poly, 即可以用一个多项式大小的电路族来判定任意 BPP 问题. 证明的方式也很简单: 只需要证明对于每个长度的输入, 都有一个 L∈BPP 的某个算法 A 中用到的随机串 r, 满足 A(x;r)=1⇔x∈L。那么关键就在于找到 r

通俗来说，只要随机算法的失误率小于 2^-n ，那么总存在固定的序列 r ，使得其能在多项式时间内运行。这个方法比枚举好的地方在于：只要固定了 r ，那么就有多项式时间算法，问题在于找到 r 不知道需要多少时间。

3. Nondeterminism(非确定性方法)

虽然 BPP 和 NP 的关系是未知的，但是我们可以证明 RP∈NP。对于 L∈RP 问题来说，可以构造一个算法 A 判定 L，且该算法没有 0-sided error，即永不可能在 x∉L 时输出 1. 那么，对于任意 y∈L，我们认为 r 是 y 的 witness，如果 A(y; r)=1. 这样一来，任意 x∉L 就每一任何 witness，方可证明:RP∈NP。

对于 BPP 的讨论会复杂一些.我们可以证明 BPP∈Σ2∩Π2。

同时，P=NP 可以推出 P=BPP，所以 NP 的“能力”更强一些，非确定性可以实现去随机化。

4. The Method of Conditional Expectations（条件期望方法）

上述三种方法：枚举，非一致性、非确定性对于所有的 BPP 是通用的，但是他们不能被确定性算法实现，而条件期望方法则是可以有效实现的，但是并不总通用的。

考虑所有使用了 m 个随机位的算法，其随机的过程都可以看作为一个深度为 m 的二叉树，每一条路径表示一个结果，在随机化算法中，我们知道大多数路径都能给定好的结果等价地，我们试图找出逐位的不错的随机序列，给定序列 (r1,r2...rn)，那么

$$\begin{aligned} P(r_1, r_2, \ldots, r_i) &\stackrel{\text{def}}{=} \Pr_{R_1, R_2, \ldots, R_m} [A(x; R_1, R_2, \ldots, R_m) \text{ is correct} \\ &\quad | R_1 = r_1, R_2 = r_2, \ldots, R_i = r_i] \\ &= \mathop{\mathrm{E}}_{R_{i+1}} [P(r_1, r_2, \ldots, r_i, R_{i+1})]. \end{aligned}$$

(See Figure 3.1.)

我们总有 P(r1,r2...rn,rn+1)≥P(r1,r2...rn)。通过最大化 rn+1，由此 P(r1,r2,...,rm)≥P(r1,r2,...,rm−1)≥·≥P(r1)≥P(Λ)≥2/3。要实现这个方法，我们需要确定性地计算 P(r1,r2...ri)，这个可能不可行。

5. Pairwise Independence（成对独立性）

(i) Pairwise Independent Hash Functions

有些情况需要从更大打范围取值获取成对独立的随机变量。比如我们想要 N=2n 个随机变量，其中每一个随机变量都是均匀分布在 {0,1}^m=[M]，那么朴素的做法是重复上述说法，每个随机变量 m 次，那么就用到了随机位 m*n=(logM)*(logN) 个。

(ii) Hash Tables

如果 M 不是关于 N 的常数，那么这个就不是 log N 的时间复杂度，我

们知道这样的序列等价于在 [M]^n 中选取样本，或者等价地，是一 hash 映射 f:[N]−>[M]。

我们可以通过构造一个这样的映射实现去随机化：构造一族映射 f:[N]−>[M]。随机取一个映射，如果这样的映射：映射的结果彼此独立且不同的元的映射不同。上述两条件等价于，任取 [N] 中两个不同的元，任取一个映射，由于其两两独立的性质，其映射到[M]上的任意两个元的概率是：

$$\forall x_1 \neq x_2 \in [N], \forall y_1, y_2 \in [M], \Pr_{H \xleftarrow{R} \mathcal{H}}[H(x_1) = y_1 \wedge H(x_2) = y_2] = \frac{1}{M^2}.$$

这样 hash 函数族被称为 explicit，如果任取 [N] 中一个元和一个函数 h，有 h(x)可以在 poly(logM,logN) 中计算，这个函数族也称为 strongly 2-universal hash functions。如下图所示：

Definition 3.22. A family of functions $\mathcal{H} = \{h : [N] \to [M]\}$ is *explicit* if given the description of h and $x \in [N]$, the value $h(x)$ can be computed in time $\text{poly}(\log N, \log M)$.

构造方法如下图所示，即对于有限域 F，若函数族 H={h(a,b):F→F}，其中 h(a,b)=ax+b 。

Construction 3.23 (pairwise independent hash functions from linear maps). Let $\mathbb{F}$ be a finite field. Define the family of functions $\mathcal{H} = \{ h_{a,b} : \mathbb{F} \to \mathbb{F}\}_{a,b \in \mathbb{F}}$ where $h_{a,b}(x) = ax + b$.

这样构造的函数族是两两独立的，此构造用了 2log|F| 个 random bit。因为存在定理：对于一个显示的两两独立的映射族，从{0,1}^N 到{0,1}^M 的随机选取的映射 H(n,m)，找到其的所需要的 random bit 是 max{m,n}+m 个，使用 Hash Tables 可以对应于这样的构造，在许多情况下，两两独立或者 k-wise 独立够用了，完全随机可能需要不切实际的存储空间。归纳总结为下图所示：

Theorem 3.26. For every $n, m \in \mathbb{N}$, there is an explicit family of pairwise independent functions $\mathcal{H}_{n,m} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ where a random function from $\mathcal{H}_{n,m}$ can be selected using $\max\{m,n\} + m$ random bits.

(iii) Randomness-Efficient Error Reduction and Sampling

假设我们有一个 BPP 算法，其失误的概率为常数，我们想要削弱其犯错

概率到 2^(-k)， 通过切比雪夫不等式，我们知道用 O(k)个随机位可以做到，如果算法本身用了 m 个随机位。那么为了削弱失误概率我们需要 O(k*m)个随机位。

用切比雪夫不等式来证明，一对独立随机变量的总和集中在其期望值周围，即形式化描述为：

Proposition 3.28 (Pairwise Independent Tail Inequality). Let $X_1, \ldots, X_t$ be pairwise independent random variables taking values in the interval $[0,1]$, let $X = (\sum_i X_i)/t$, and $\mu = \mathrm{E}[X]$. Then

$$\Pr[|X - \mu| \geq \varepsilon] \leq \frac{1}{t\varepsilon^2}.$$

这个定理告诉我们，如果我们使用 t=O(2^k)配对独立重复，我们可以将 BPP 算法的错误概率从 1/3 降低到 2^-k。 如果原始 BPP 算法使用 m 个随机比特，那么我们可以通过选择 h ： {0,1}^(k+O(1)) → {0,1}^m，从一个成对独立的族中随机抽取，并对所有 x∈{0,1}^(k+O(1))，使用抛硬币 h(x)运行算法。这需要 m+max{m,k+O(1)}=O(m+k)个随机比特。即如下图所示：

	Number of Repetitions	Number of Random Bits
Independent Repetitions	$O(k)$	$O(km)$
Pairwise Independent Repetitions	$O(2^k)$	$O(m+k)$

(iv) k-wise Independence

刚刚提到的来两两独立的构造自然地可以拓展到 k-wise 独立，函数族的定义类似地延拓即可；为了证明唯一性，假设我们有两个度数最多为 k-1 的多项式 h 和 g， $h(x) = \sum_{i=1}^{k} y_i \cdot \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$. 。使得 i=1,...,k 的 h(xi)=g(xi)。 那么 h-g 至少有 k 个根，因此必须是零多项式。综上在 k-wise 独立的情景下就是一个 k-1 次多项式；从而有以下推论，从 $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ 中选择一个随机函数需要 k*max{n,m}个随机位，评估 H 的一个函数需要 ploy(n,m,k)时间。

姓名：钱泽凯

学号：1190202011