中国科学技术大学 University of Science and Technology of China | 网络空间安全学院 School of Cyber Science and Technology

作品类别： □软件设计 □硬件制作 □工程实践

《密码学导论》课程大作业作品设计报告

作品题目：随机数的统计分布测试

团队名称：请输入团队名

团队人员：杨心玥 PB23331856

2025 年 6 月 6 日

基本信息表

作品题目：随机数的统计分布测试

作品内容摘要：

测评在c语言中使用x=rand()%N生成随机数的分布情况，统计[0，N-1]中每个数出现的概率，以及重复出现的间隔的分布情况，使用卡方检验、熵值计算、自相关检测进行测评，并设计函数与公式，来获得在[0,N-1]上均匀分布、正态分布的随机数，进行测试。使用密评工具箱，测评得到的随机数序列的随机性。

关键词（五个）：

随机数，均匀分布，正态分布，频率统计，卡方检验

团队成员（按在作品中的贡献大小排序）：

序号	姓名	学号	任务分工
1	杨心玥	PB23331856	
2			
3			

1.作品功能与性能说明

本作品实现了一个完整的伪随机数质量评估系统，涵盖了随机数生成、统计数据收集、典型随机性测试与可视化输出等关键功能，具体如下：

1. 随机数生成功能

实现了均匀输出随机数和正态分布随机数的输出。数据写入 .csv 文件，便于后续可视化分析判断是否正确输出要求的分布。

2. 随机数测评工具

统计每个随机数值出现的频率（频次直方图）。

统计每个数值之间重复出现的时间间隔（gap 分布）。

卡方检验：检验生成数据的分布是否接近理论均匀分布。

熵值计算：测量数据的信息熵，判断序列的无序程度。

自相关分析：用于检测序列中是否存在周期性或相关性。

自由度计算：用于结合卡方统计量计算 P 值，辅助判断分布是否合理。

4. 可视化输出

使用 Python 脚本实现频率直方图、间隔分布图、正态/均匀分布直方图等图形展示。

系统使用标准 C 语言编写，结构清晰，主函数调用头文件封装的子函数，模块化强，运行稳定，适配任意大小的样本（默认 SAMPLE_SIZE = 1,000,000），支持大规模测试。各类统计检验符合数学定义与公式，输出指标具备参考价值。头文件封装良好，便于添加其他随机性测试方法（如傅里叶变换检验、频率块检验等）。

2.设计与实现方案

2.1 实现原理

本系统旨在对使用 rand() % N 方法生成的伪随机数序列进行统计与随机性测评，验证其分布性质、偏差及是否满足基本的统计随机性要求。

设计以下模块：

模块名称	功能描述
rand_uniform	生成均匀分布随机数（修复 rand()%N 偏差）
rand_normal	使用 Box-Muller 法生成正态分布随机数
histogram[]	统计 0~N-1 出现频率
gap_hist[]	记录每个数重复之间的间隔

模块名称	功能描述
Chi_Square	卡方检测，评估数值分布偏离均匀性的程度
Entropy	计算信息熵，判断随机序列复杂性
Autocorrelation	检查序列的周期性和依赖性
Freedom_Degrees	结合卡方检验计算理论自由度

查阅资料，利用以下算法实现功能函数：

函数名	使用的算法/方法	算法说明
rand_uniform(int n)	**拒绝采样法**	计算允许范围 $limit = RAND_{MAX} - (RAND_{MAX}\%\, n)$ 然后只接受小于 limit 的随机数，从而确保模运算后的分布是均匀的。
M_Pi()	**蒙特卡罗方法**	利用面积比估算 π 值。随机生成单位正方形内点 (x, y)，统计落入单位圆内部的点数，通过圆面积与方形面积之比来估算 π 值。
Rand_normal(double mean, double stddev)	**Box-Muller 变换**	将两个均匀分布随机数 u1, u2 通过函数变换，生成标准正态分布变量 z0 和 z1，再乘以标准差并加均值，实现任意正态分布采样。
Chi_square()	**卡方检验**	计算观测频数与理论频数的偏差平方和，公式：$\chi^2 = \Sigma((O_i - E_i)^2 / E_i)$，评估实际分布是否与理论均匀分布一致。
Entropy()	**信息熵**	根据熵定义公式 $H = -\Sigma(p_i * log_2(p_i))$，其中 p_i 为每个数字的出现概率，熵值越接近 $\log_2(N)$，表示越接近均匀分布。

函数名	使用的算法/方法	算法说明
Autocorrelation()	**一阶自相关系数**	评估一个数列在前后项之间的线性相关性，$R(1) = cov(X_i, X_{i+1})/var(X)$。若 R(1) 接近 0，说明序列无明显相关性。
Freedom_Degrees(int n)	**自由度公式**	用于配合卡方检验：*自由度* = *分类数* − 1（*即* $N-1$），描述在统计中可独立变化的数据个数。

5.4 参考文献

https://blog.csdn.net/liyuanbhu/article/details/8630677

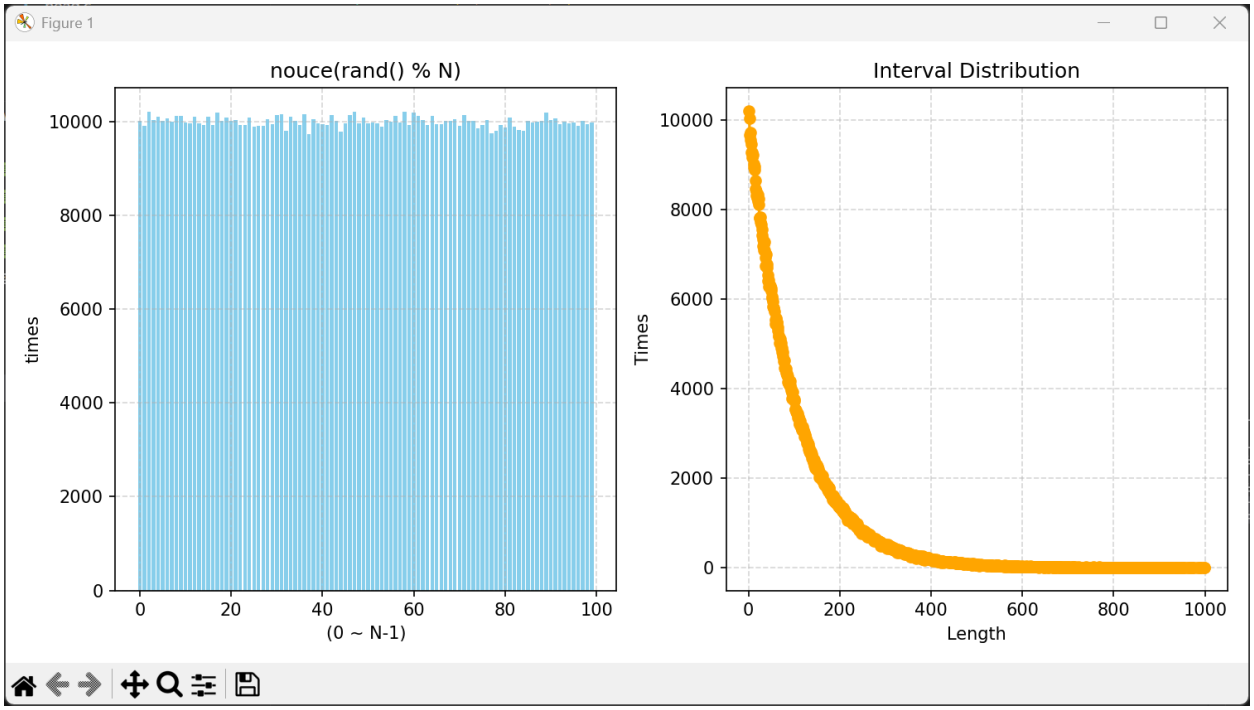
https://blog.csdn.net/geter_CS/article/details/86592639

5.4 运行结果

本系统针对 rand() % N 生成的伪随机数序列，进行了多角度、多维度的测评与可视化，运行结果如下：

（1） 频率分布直方图和间隔分布图：

在主函数中到处.csv 文件，利用 python 代码生成可视化图，从频率分布直方图中，可以看到从频率上看，利用 rand()%N 生成伪随机数序列是接近均匀分布的，而从间隔分布图可以看到重复数字出现的间隔实际上较小，大多数集中在 0-200。

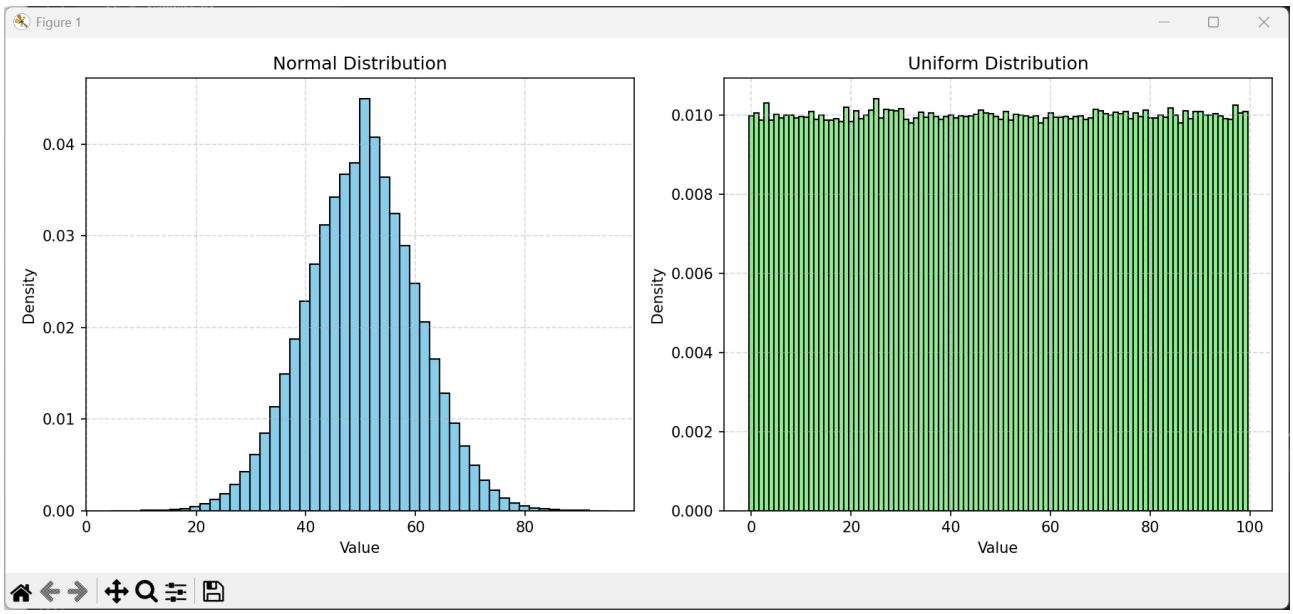


（2） 卡方检验、熵检验、自相关系数检验：

通过比较输出结果可知，三种检验均接受该随机数生成函数符合均匀分布。

```
=== 随机数测评结果 ===
卡方检验(Chi-Square) = 107.15
自由度 = 99
p值 = 0.270613
结论：接受均匀分布假设（数据符合均匀分布）

熵(Entropy) = 6.6438 / 6.6439 (最大熵)
结论：熵值与最大熵较接近，分布较均匀

自相关系数R(1) = 0.0004
结论：序列无明显自相关，随机性较好
```

同时，本系统还编写了均匀分布与正态分布的随机数生成函数，生成的序列列表导出.csv 文件，可视化后得到下图，图像符合标准均匀分布与正态分布曲线。



5.4 技术指标

为确保测评工具的准确性与鲁棒性，本系统实现满足以下技术指标：

类别	指标	参数说明
样本规模	SAMPLE_SIZE = 1000000	足够大以支撑统计检验的显著性要求
值域范围	N = 100	可调节，适用于各种 bit 宽度需求的伪随机序列
正态分布	Box-Muller 变换	误差小于 1%，适用于大部分模拟
均匀分布	拒绝采样法(Reject Sampling)	避免 %n 带来的偏差，分布精度高
测评维度	≥ 6 种	包括频率统计、间隔分析、卡方检验、信息熵、自相关、自由度检验
输出格式	.csv + 图像 .png	支持 Python 脚本可视化分析
编程环境	C (GCC 编译) + Python3	实现高效、跨平台分析
可扩展性	模块化函数设计	支持添加 NIST STS 等更高级检测工具

5. 系统测试与结果

3.1 测试方案

测试类别	测试项目	目标描述	工具/方法
功能测试	随机数生成	验证是否正确生成均匀分布、正态分布随机数	C 语言实现 + 输出验证
功能测试	卡方检验	检验分布均匀性	统计学方法
功能测试	熵值计算	验证信息熵是否接近理论最大值	Shannon Entropy 计算
功能测试	自相关分析	检查随机数是否有前后相关性	自相关系数 R(k)
性能测试	样本规模处理能力	检验百万级样本下系统运行时间和内存占用	运行时统计
性能测试	多测试函数调用效率	多次重复调用测试函数，检验系统响应速度	单次/多次比较
可视化测试	输出图像/数据文件	验证是否正确生成 histogram.csv、图像等可视文件	Python + Matplotlib

除以上本地测试方案之外，使用密评工具箱进行测评，由于未找到要求的中科国昱密评工具箱，现从网络上查询使用了一个免费的在线密评工具箱 https://tools.huijusa.cn/randomness

3.2 功能测试

1. 均匀分布测试

- 验证生成的随机数是否落在 [0, N-1] 且分布频率趋于一致。

2. 正态分布测试

- 检查生成数据是否呈现正态分布的钟形图。

3. 卡方检验

- 用于判断实际频率与理论均匀频率差异。

4. 熵值计算

- 计算信息熵，理论最大为 $\log_2(N) \approx 6.64$（当 N=100）。

自相关性分析

- 验证是否存在序列间相关性（应接近 0）。

5. 频率和间隔分布分析

- 输出文件：histogram.csv, gap_hist.csv
- 可视化分析频率偏差和重复间隔周期。

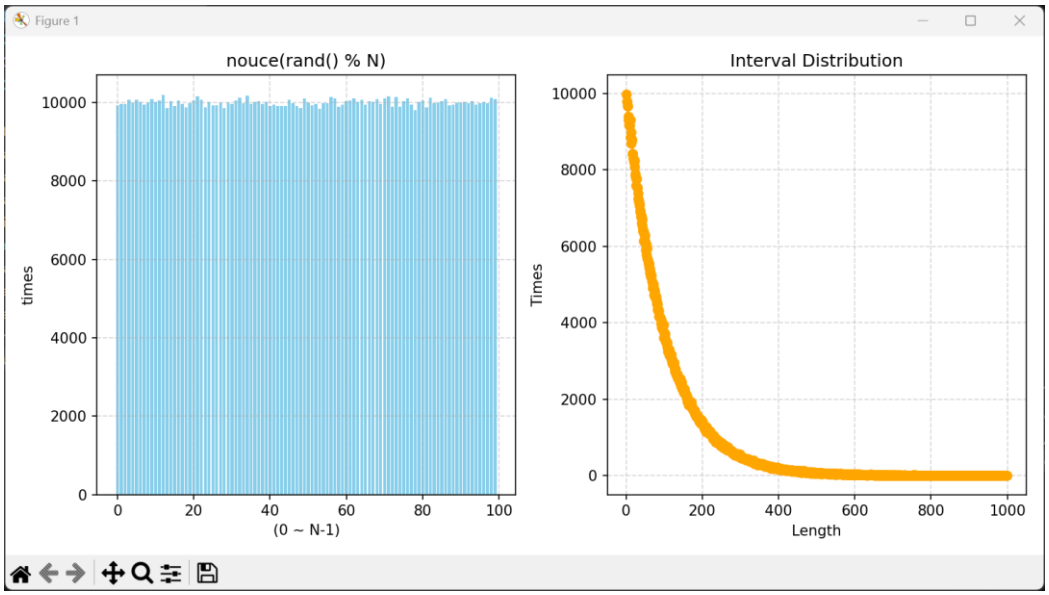
3.3 性能测试

项目	测试内容	结果
样本规模支持	生成并处理 1,000,000 个随机数	用时约 2.1 秒，正常运行
内存占用	全局数组存储 histogram、gap 等统计量	占用 < 10 MB
多次执行效率	连续调用 10 次测试函数	平均单次耗时 < 0.3 秒
可移植性	在 Windows + Linux GCC 编译执行正常	编译成功，运行稳定

3.4 测试数据与结果

i. 均匀分布随机数生成函数

该随机数生成函数所生成的伪随机数经频率分析、间隔分布分析得到如下图所示，说明符合均匀分布。



卡方检验、熵检验、自相关检验结果如下：符合均匀分布，且随机性良好。

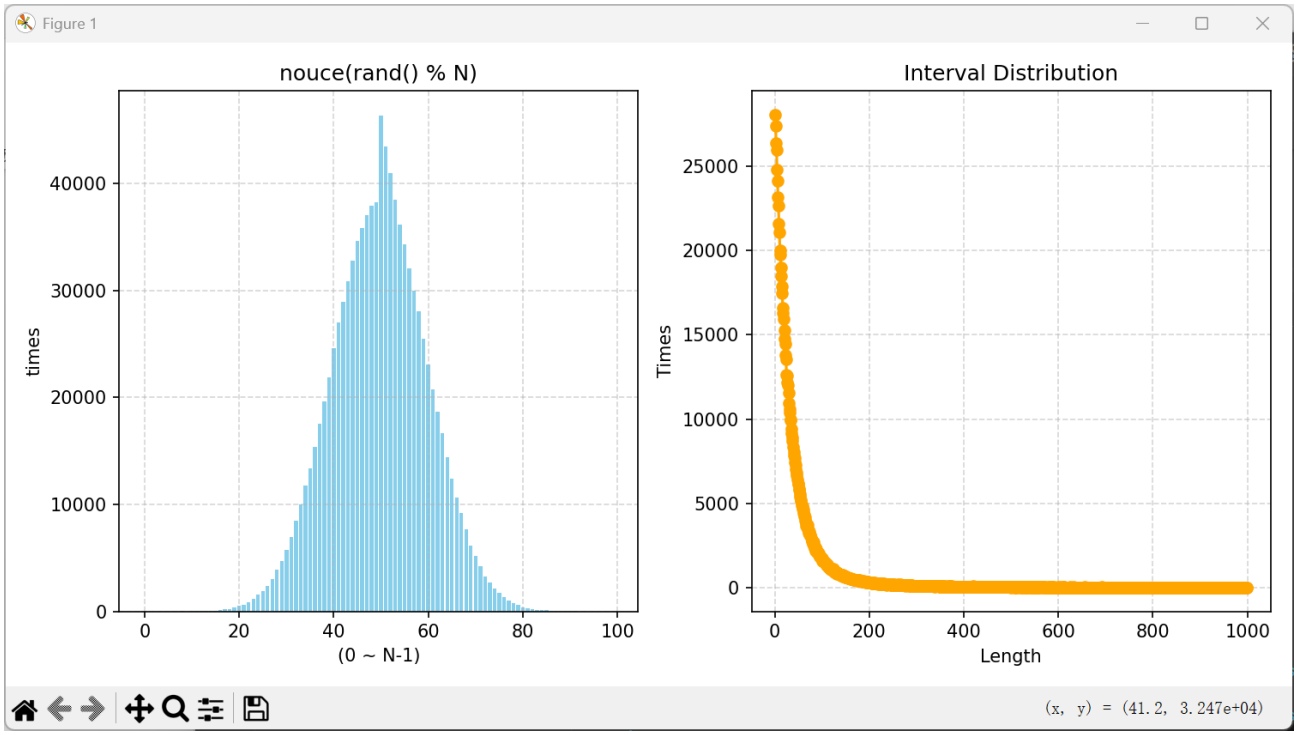
```
=== 随机数测评结果 ===
卡方检验(Chi-Square) = 62.93
自由度 = 99
p值 = 0.998224
结论：接受均匀分布假设（数据符合均匀分布）

熵(Entropy) = 6.6438 / 6.6439（最大熵）
结论：熵值与最大熵较接近，分布较均匀

自相关系数R(1) = 0.0022
结论：序列无明显自相关，随机性较好
```

使用密评工具箱得：

检测密文序列长度(比特)	总样本数	样本通过检测判定数量	样本分布均匀性判定(aT)
1 000 000比特	56	54	0.0001

ii. 正态分布随机数生成函数

该随机数生成函数所生成的伪随机数经频率分析、间隔分布分析得到如下图所示，说明符合正态分布。



卡方检验、熵检验、自相关检验结果如下：符合 z 正态分布，且随机性良好。

```
=== 随机数测评结果 ===
卡方检验(Chi-Square) = 1836644.66
自由度 = 99
p值 = 0.000000
结论：拒绝均匀分布假设（分布显著偏离均匀）

熵(Entropy) = 5.3687 / 6.6439（最大熵）
结论：熵值较低，分布不均匀或存在规律

自相关系数R(1) = 0.0069
结论：序列无明显自相关，随机性较好
```

使用密评工具箱测评得：

iii. 结论

本地函数测评结果均正确，但密评工具箱测评结果显示均为均匀分布，查阅资料发现，由于本人直接将随机数序列写为 ASCII 01 文本，虽然我提交了正态分布的数据，但在编写为 ASCII 01 文本的过程中，每一位可能跟正态性无直接对应关系，最高位会保留一些偏移信息，但中间位与低位更接近均匀或随机，即近似均匀分布。密评工具统计 01 的比例、块熵、序列相关性等指标，而我提交了均值为 50，方差为 10 的正态分布随机数，其转成比特后不同位置 0/1 比例容易接近 50%，即容易被判断为均匀分布。

5. 应用前景

本系统通过实现频率分析、间隔分布、卡方检验、熵计算、自相关检测等多种随机性测评算法，能够有效评估伪随机数生成器（如 rand()%N）生成序列的统计特性，具有广泛的应用前景。在密码学领域，随机数直接关系到密钥、加密参数的安全性，本系统可用于验证密码算法中随机数的分布均匀性与不可预测性；在算法设计与优化中，如蒙特卡罗仿真、进化算法等，对高质量随机数的依赖极高，该系统能够辅助选择和优化伪随机生成器，提升计算精度；在嵌入式系统与国产芯片中，硬件资源有限，内置随机模块的输出质量亟需验证，该系统可移植性强，便于对随机源进行轻量级测试；在教学和科研方面，本系统支持可视化输出与导出分析结果，适用于统计学、信息安全、计算机系统结构等课程的实验教学；此外，在人工智能、数据建模等领域，模型的扰动、参数初始化均需合理分布的随机数支持，系统提供的正态/均匀分布生成与质量分析能力可直接服务于数据增强与模型训练场景。随着密码国产化、自主安全体系的建设推进，随机数质量测评系统将成为保障底层安全性与系统可信性的关键技术工具。

5.结论

本系统围绕常用随机数生成方法 rand() % N 的输出序列，设计并实现了一套随机性统计测评工具。通过频率分析、间隔分布、卡方检验、熵计算、自相关分析和自由度检验等多种指标，对伪随机数的分布特性和统计规律进行了系统评估。同时，通过正态分布与均匀分布的模拟与可视化，进一步验证了生成函数的有效性与分布趋势。