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

《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

作品类别： 软件设计

团队名称：

团队人员： 李友好PB23030784

2025年6月8日

1 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： 以下是程序员经常使用的产生0到N-1之间随机整数的方法“x=rand()%N; //以C语言为例”针对于此本作业实现了以下功能（均采用交互功能通过询问获得输入）：
① 按“x=rand()%N”随机生成后统计每个数的出现概率
② 按“x=rand()%N”随机生成后统计每个数重复出现的间隔的分布情况
③ 对“x=rand()%N”方法进行熵值测评
④ 对“x=rand()%N”方法进行游程测评
⑤ 对“x=rand()%N”方法进行自相关测评
⑥ 对“x=rand()%N”方法进行频率测评
⑦ 对“x=rand()%N”方法进行累加测评
⑧ 设计公式或函数，获得均匀分布的随机数，正态分布的随机数，并测试

关键词（五个）： 随机数，统计分布，测评,正态分布，均匀分布
团队成员（按在作品中的贡献大小排序）： 李友好PB23030784

2 作品功能与性能说明

2.1 作品功能说明

1. **随机数频率统计 (count_frequency)**

功能：生成指定范围内的随机整数，统计每个随机数出现的频率。用户需输入范围 N 和生成随机数的次数 count_times。

实现方式：使用 rand()%N 生成随机数，并通过数组 times 记录每个随机数出现的次数，最后计算并输出每个随机数的频率。

2. **重复间隔统计 (count_repetition_intervals)**

功能：统计指定范围内随机数重复出现的间隔分布。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：使用二维数组 intervals 记录每个随机数重复出现的间隔次数，通过 last_pos 数组记录每个随机数上一次出现的位置。最后输出每个随机数的间隔分布情况。

3. **熵测试 (entropy_test)**

功能：对生成的随机数进行熵测试，评估随机数的随机性。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：生成随机数并统计每个数的出现次数，计算实际熵值和理想熵值，根据实际熵值是否达到理想熵值的 95% 判断测试是否通过。

4. **游程测试 (runs_test)**

功能：对生成的随机数进行游程测试，评估随机数的随机性。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：将随机数转换为二进制序列，统计游程数量，计算期望游程数、方差和标准差，通过 z-score 判断测试是否通过。

5. **自相关测试 (autocorrelation_test)**

功能：对生成的随机数进行自相关测试，评估随机数的独立性。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：计算随机数的均值、自相关系数和标准误差，通过 z-score 判断测试是否通过。

6. **频率测试 (frequency_test)**

功能：对生成的随机数进行频率测试，评估随机数的均匀分布性。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：统计每个随机数的出现次数，计算卡方统计量和自由度，通过 p-value 判断测试是否通过。

7. **累积和测试 (cumulative_sums_test)**

功能：对生成的随机数进行累积和测试，评估随机数的随机性。用户输入范围 N 和生成随机数的次数 count_times。

实现方式：将随机数转换为标准化值，计算正向和反向累积和的最大值，计算 z-score 和 p-value，根据 p-value 判断测试是否通过。

8. **随机数生成函数**

- 均匀分布随机数 (uniform_random)：生成 0 到 $N-1$ 之间的均匀分布随机数。
- 正态分布随机数 (normal_random)：使用 Box - Muller 变换生成指定均值和标准差的正态分布随机数。

9. **统计计算函数**

- 均值计算 (mean)：计算数组中有效元素的均值。
- 标准差计算 (stddev)：计算数组的标准差。
- 中位数计算 (median)：计算数组的中位数。

2.2 作品性能说明

1. **时间复杂度**

- 随机数生成：使用 rand() 函数生成随机数，时间复杂度为 $O(1)$。
- 频率统计和间隔统计：需要遍历生成的随机数，时间复杂度为 $O(\text{count\_times})$。
- 熵测试、游程测试、自相关测试、频率测试和累积和测试：主要操作是遍历随机数和进行统计计算，时间复杂度为 $O(\text{count\_times})$。
- 中位数计算：使用冒泡排序，时间复杂度为 $O(n^2)$，其中 n 是数组的长度。

2. **空间复杂度**

- 频率统计和间隔统计：需要使用数组记录每个随机数的出现次数和间隔次数，空间复杂度为 $O(N)$。
- 熵测试、游程测试、自相关测试、频率测试和累积和测试：需要使用数组存储随机数和统计结果，空间复杂度为 $O(\text{count\_times})$。
- 中位数计算：需要对数组进行排序，空间复杂度为 $O(1)$。

3 设计与实现方案

3.1 2.1 实现原理

① **统计随机数出现概率（count_frequency）**

原理：通过 rand()%N 生成 $[0, N-1]$ 区间的随机整数，利用数组 times 记录每个数的出现次数。

生成指定次数的随机数后，将每个数的出现次数除以总次数，得到其出现概率。核心逻辑是频率统计，通过直接计数和比例计算，直观反映每个数值在随机序列中的分布密度。

② 统计随机数重复间隔分布（count_repetition_intervals）

原理：使用数组 last_pos 记录每个数最后一次出现的位置。每次生成随机数 x 时，若 x 非首次出现，则计算当前位置与上一次位置的间隔 $\text{gap} = i - \text{last\_pos}[x] - 1$，并通过二维数组 $\text{intervals}[x][\text{gap}]$ 统计间隔 gap 的出现次数。最终输出每个数的间隔分布频率，用于分析随机数重复出现的时间间隔规律，评估序列的随机性是否符合均匀间隔预期。

③ 熵值测评（entropy_test）

原理：根据信息熵公式 $H = -\sum(p_i \log_2 p_i)$，计算随机序列的实际熵值。其中 p_i 是每个数的出现概率。理想情况下，均匀分布的熵值为 $\log_2 N$。通过比较实际熵值与理想熵值的 95% 阈值，判断序列的随机性是否接近理论最优（均匀分布）。熵值越高，序列的不确定性越强，随机性越好。

④ 游程测评（runs_test）

原理：将随机数转换为二进制序列（以中位数为阈值，大于等于阈值为 1，否则为 0），统计连续相同数字的“游程” 数量。根据理论公式计算期望游程数和标准差，通过 z-score 检验实际游程数与期望值的偏离程度。若 $|z| < 1.96$（95% 置信区间），则认为序列的游程分布符合随机序列的统计特性。

⑤ 自相关测评（autocorrelation_test）

原理：计算随机序列中相邻数据的自相关系数，衡量前后数据的依赖程度。理想随机序列的自相关系数应接近 0。通过计算均值、自相关系数和标准误差，得到 z-score。若 $|z| < 1.96$，说明序列中相邻数据无显著相关性，符合随机性要求。

⑥ 频率测评（frequency_test）

原理：使用卡方检验评估随机数的频率分布是否均匀。计算每个数的实际出现次数与理论期望次数（总次数 / N）的差异，构造卡方统计量 $\chi^2 = \sum \frac{(O-E)^2}{E}$。根据自由度 $N-1$ 和 p-value 判断：若 $p > 0.05$，则接受“均匀分布” 假设，认为序列频率分布无显著异常。

⑦ 累加测评（cumulative_sums_test）

原理：将随机数转换为标准化值（大于等于中位数为 1，否则为 -1），计算正向和反向累加和的最大值。通过标准化处理（除以 $\sqrt{n}$）得到统计量 z，并近似计算 p-value。若 $p > 0.01$，说明累加和的波动在随机序列的合理范围内，未显示非随机性趋势。

⑧ 均匀分布与正态分布随机数生成及测试（last_part_test）

原理：

均匀分布：通过 $\text{uniform\_random}(N) = \frac{\text{rand()}}{\text{RAND\_MAX}} \times N$ 生成 $[0, N)$ 区间的均匀随机数，利用卡方检验和直方图验证其分布均匀性。

正态分布：使用 Box-Muller 变换，通过两个均匀随机数 $u1, u2$ 生成标准正态随机数 $Z = \sqrt{-2\ln u1}\cos(2\pi u2)$，再通过 $X = \mu + \sigma Z$ 得到指定均值和标准差的正态随机数。通过 Lilliefors 检验和直方图验证其正态性。

测试逻辑：计算均值、标准差、中位数等统计量，与理论值对比，并通过拟合优度检验判断生成的随机数是否符合目标分布（均匀或正态）。

3.2 2.2 参考文献

3.3 2.3 运行结果

实现菜单功能，可以进行各部分正常运行

3.4 2.4 技术指标

4 系统测试与结果

4.1 3.1 测试方案

直接运行“单人作品.cpp”即可

4.2 3.2 功能测试

各部分功能正常

4.3 3.3 性能测试

运行速度较快

4.4 3.4 测试数据与结果

测试结果正常

5 应用前景

可以用于评测一些随机数生成方法

6 结论

本作业实现了单人作品3随机数的统计分布测试，并且能够正常运行，采用了多种测评工具进行测评，能正常完成要求，但也存在一些局限，如：测评方法在大小样本的对待敏感度有差异

中科国昱密评工具箱未找到有效版本，经过询问，助教老师说可不用做。