从生日问题理解碰撞事件发生的概率

谢牧航 2022211363　　杨百川 2022211362　　王宏伟 2022211377

论文摘要

本文深入探讨了生日问题，这是概率论中一个众所周知的一个悖论。我们的研究分为理论计算和基于 Python 的仿真模拟，全面分析了这个问题的各个方面。首先，我们关注至少两个人同日生日的概率，使用理论计算和仿真模拟两种方法探索这一问题。这种双重方法允许我们通过实证数据来验证理论预测。随后，我们将研究扩展到至少三个人同日生日的概率和至少有两个人的生日相差不超过一天的概率。同样，我们运用仿真模型为理论基础提供了实际视角。这部分研究不仅加强了对核心生日问题的理解，还为其更广泛的应用奠定了基础。

此外，本文还探讨了生日问题在密码学中的应用，特别是 Pollard Rho 算法和哈希生日攻击。我们详细阐述了这些方法的原理和理论计算，进一步展示了生日问题在解决实际安全问题中的重要性。同时，我们还讨论了生日攻击在解决离散对数问题中的应用，包括随机解法和大步小步算法（Baby-Step Giant-Step, BSGS）。这些探讨不仅凸显了生日问题在理论和实际应用中的双重价值，也为进一步的研究提供了坚实的基础。

目录

1 基础

我们首先讨论至少两个人生日相同的概率。

1.1 理论计算

设一共有 n 天，k 个人生日互不相同为事件 A, 则事件 A 的概率为

$$P(A)=\prod_{i=0}^{k-1}\frac{n-i}{n}$$

至少有两个人生日相同的概率为 $P\left(\overline{A}\right)=1-P(A)$。根据题意可知 $P\left(\overline{A}\right)\geq\frac{1}{2}$, 那么就有

$$P(A)=\prod_{i=0}^{k-1}\frac{n-i}{n}\leq\frac{1}{2}$$

事实上到这一步之后只能通过从小到大枚举 k 的值来求解满足的 k。当 $n=365$，枚举得到满足的最小 k 为 $k=23$，此时 $P(A)\approx0.5073$。

数学家保罗 · 哈莫斯给出了一种使用不等式的近似解法如下：

由不等式 $1+x\leq\mathrm{e}^x$ 可得

$$P(A)\leq\prod_{i=1}^{k-1}\exp\left(-\frac{i}{n}\right)=\exp\left(-\frac{k(k-1)}{2n}\right)$$

因此

$$\exp\left(-\frac{k(k-1)}{2n}\right)\leq\frac{1}{2}\Longrightarrow P(A)\leq\frac{1}{2}$$

将 $n=365$ 代入，解得 $k\geq23$。所以一个房间中至少 23 人，使其中两个人生日相同的概率达到 50%；当 $k>56$，$n=365$ 时，出现两个人同一天生日的概率将大于 99%。

但是需要注意，这种方法只显示**至少**超过 23 人就能保证平等机会下的生日匹配。但是无法借此计算过程确定 n＝22 是否能让概率过半。

但是这个式子仍然是有意义的，在泛化的两人生日问题中：给定从符合离散均匀分布的区间 $[1,n]$ 随机取出 k 个整数，至少 2 个数字相同的概率 $p(k;n)$有多大？当 n 很大时，我们仍可以使用这个近似的概率来进行估计：$p(k;n)\approx1-\exp\left(-\frac{k(k-1)}{2n}\right)\approx1-\exp\left(-\frac{k^2}{2n}\right)$。同时我们可以利用这个式子来进行反向估计，给定 p 来找到最小能满足的 k：$k(p;n)\approx\sqrt{2n\ln\left(\frac{1}{1-p}\right)}$。

1.2 仿真模拟

使用 Python 仿真模拟来绘制频率变化曲线。（Python 源代码见附件）

1. 对于每一个 $n=k$（$k=2,3,...,100$），进行 100000 次随机模拟。
2. 在每次模拟中，我们会随机生成 n 个介于 1 和 365 之间的整数，用以代表 n 个人的生日。
3. 我们将检查是否存在至少一对相同的生日。
4. 记录每次模拟结果，统计出现生日相同的频率。
5. 绘制 $n=k$ 和生日相同的累积频率之间的关系图。

使用 `matplotlib` 绘制出的图像如下：

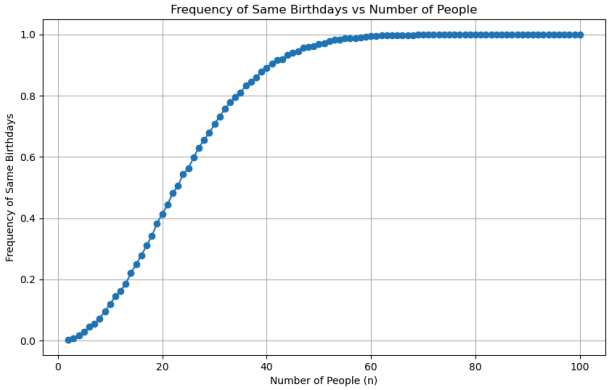


图 1　至少两个人生日在同一天的频率

以上是 100 次试验中出现生日相同的累积频率变化曲线。横坐标表示人数 n（从 2 到 100），纵坐标表示出现至少一对生日相同的频率。

从图中可以明显看出，随着人数 n 的增加，出现至少一对生日相同的频率也逐渐增加。这与生日问题的理论预测是一致的。

在人数达到 23 人以上时，至少两人生日相同的频率已大于 50%。在人数达到 60 人以上时，至少两人生日相同的频率已大于 99%。

2 拓展

本章我们会探讨更多情况下的生日问题。

2.1 至少三个人生日在一天的概率

2.1.1 理论计算

要计算至少三个人生日在一天的概率, 依然可以反向思考, 用总体减去每个人生日互不相同的概率，再减去恰有两人生日相同的概率即可。

假设这个班有 n 个人（根据现实情况，我们只考虑 n 是不超过 365 的正整数），那么每个人生日互不相同的概率为

$$P_1 = \frac{\prod_{i=0}^{n-1}(365-i)}{365^n}$$

而恰有两个人生日相同的概率 P_2 要分有多少“生日相同 2 人组”。当 $n>2$，如果有 k 组，先从 n 个人中选 $2\times k$ 个人，有 C_n^{2k} 种情况；再两两分组，$(2*k-1)!!$ 种情况；之后可以视两人组为一个人，即班中有 $n-k$ 个人，每个人生日不能再相同。那么

$$P_{2k} = C_n^{2k} \times (2*k-1)!! \times \frac{\prod_{i=0}^{n-k-1}(365-i)}{365^n}, \quad k=1,2,\cdots\lfloor n/2\rfloor$$

所以

$$P_2 = \sum_{k=1}^{\lfloor n/2\rfloor} C_n^{2k} \times (2*k-1)!! \times \frac{\prod_{i=0}^{n-k-1}(365-i)}{365^n}$$

所以

$$P = 1-P_1-P_2 = 1-\frac{\prod_{i=0}^{n-1}(365-i)}{365^n} - \sum_{k=1}^{\lfloor n/2\rfloor} C_n^{2k} \times (2*k-1)!! \times \frac{\prod_{i=0}^{n-k-1}(365-i)}{365^n}$$

对 n 进行遍历找到最小的 n 使得 $P>0.5$，就求得了需要多少人才可以保证至少 3 个人在同一天的生日概率不小于 50%。使用 Python 进行编程计算（源代码见附件），我们得到以下概率表：

班级人数	三人及以上生日相同出现频率	班级人数	三人及以上生日相同出现频率	班级人数	三人及以上生日相同出现频率
50	0.1252	68	0.2858	86	0.4884
51	0.1321	69	0.2960	87	0.5003
52	0.1384	70	0.3077	88	0.5086
53	0.1491	71	0.3204	89	0.5190
54	0.1571	72	0.3330	90	0.5264
55	0.1633	73	0.3506	91	0.5512
56	0.1725	74	0.3541	92	0.5587
57	0.1764	75	0.3594	93	0.5725
58	0.1829	76	0.3670	94	0.5696
59	0.1995	77	0.3912	95	0.5894
60	0.2088	78	0.3953	96	0.6047
61	0.2068	79	0.4088	97	0.6064
62	0.2246	80	0.4216	98	0.6233
63	0.2450	81	0.4287	99	0.6323
64	0.2346	82	0.4367	100	0.6448
65	0.2563	83	0.4510		
66	0.2635	84	0.4665		
67	0.2724	85	0.4786		

于是可知当班级人数超过 87 人，出现三个人生日相同的概率超过 50%。

2.1.2 仿真模拟

枚举一个班总人数 n，随机生成 n 个介于 1~365 的数字作为这 n 个人的生日，检查这 n 个数字是否出现有某个数字出现三次或更多。

(Python 源代码见附件)

进行重复随机生成 100000 次得到以下仿真结果：

班级人数	三人及以上生日相同出现频率	班级人数	三人及以上生日相同出现频率	班级人数	三人及以上生日相同出现频率
50	0.1254	59	0.1974	68	0.2858
51	0.1320	60	0.2082	69	0.2960
52	0.1411	61	0.2157	70	0.3077
53	0.1477	62	0.2266	71	0.3169
54	0.1562	63	0.2361	72	0.3277
55	0.1648	64	0.2461	73	0.3384
56	0.1703	65	0.2562	74	0.3494
57	0.1804	66	0.2643	75	0.3645
58	0.1895	67	0.2737	76	0.3730
77	0.3837	86	0.4863	95	0.5920
78	0.3942	87	0.4969	96	0.6028
79	0.4053	88	0.5112	97	0.6129
80	0.4215	89	0.5218	98	0.6252
81	0.4271	90	0.5333	99	0.6364
82	0.4383	91	0.5469	100	0.6441
83	0.4498	92	0.5585		
84	0.4635	93	0.5680		
85	0.4758	94	0.5814		

画图如下：

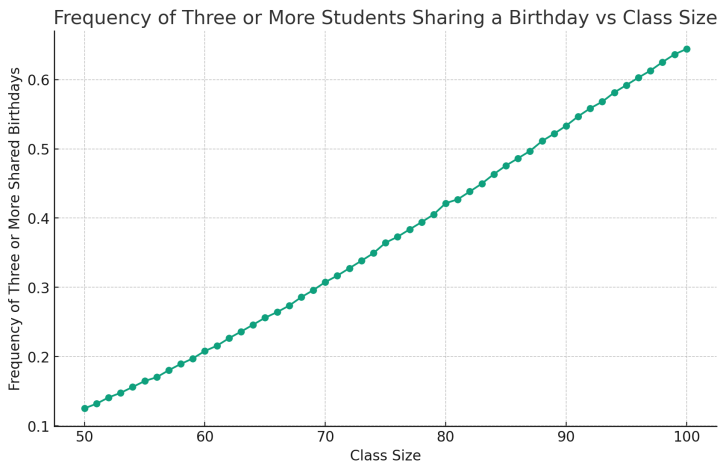


图 2　至少三个人生日在一天的频率

仿真结果和理论计算结果基本一致。

2.2 至少两个人的生日相差不过一天的概率

2.2.1 理论计算

要计算至少两个人的生日相差不过 1 天的概率，依然可以反向思考，用总体减去每个人生日互不相邻的概率。这里我们认为 12 月 31 日和 1 月 1 日是两个相邻的日期，则每个人生日互不相邻的概率可以用环式不相邻的方法计算。假设这个班有 n 个人（$n < 183$，当 $n \geq 183$ 所求概率为 0）将日期视为围成一圈的 365 个小球，取其中 n 个球，但所取的球位置不相邻，共有 $C_{365-n-1}^{n-1} + C_{365-n}^{n}$ 种取法（此式求解见注释），再将球和人对应起来，由于球是相同的但人是不同的，故产生$n!$ 种对应组合。不加限制的生日可能依然是 365^n 种，综上

$$P = 1 - \frac{n! \times (C_{365-n-1}^{n-1} + C_{365-n}^{n})}{365^n}$$

使用 Python 进行编程计算（源代码见附件），我们得到以下概率表：

班级人数	至少两个人的生日相差不过一天的概率	班级人数	至少两个人的生日相差不过一天的概率
1	0	11	0.3706
2	0.0082	12	0.4269
3	0.0245	13	0.4829
4	0.0485	14	0.5375
5	0.0797	15	0.5901
6	0.1174	16	0.6399
7	0.1607	17	0.6866
8	0.2087	18	0.7297
9	0.2604	19	0.7690
10	0.3147	20	0.8045

于是可知当班级人数超过 14 人，出现 2 个人的生日相差不过 1 天的概率的概率超过 50%。

综上所述，我们可以看到如果加强生日问题的条件至三人，发生碰撞概率大于 50% 也只需要 88 人，若放宽条件到相邻，则只需要 14 人，碰撞发生概率之高使得我们有更多的思考。

2.2.2 仿真模拟

枚举一个班总人数 n，随机生成 n 个介于 1~365 的数字作为这 n 个人的生日，检查这 n 个数字是否出现有两个相邻数字同时出现（1 和 365 认为是相邻）的情况。

（Python 源代码见附件）

进行重复随机生成 100000 次得到以下仿真结果：

班级人数	至少两个人的生日相差不过一天的概率	班级人数	至少两个人的生日相差不过一天的概率
1	0	11	0.3719
2	0.0081	12	0.4257
3	0.0247	13	0.4833
4	0.0474	14	0.5359
5	0.0799	15	0.5913
6	0.1190	16	0.6400
7	0.1635	17	0.6870
8	0.2083	18	0.7299
9	0.2593	19	0.7699
10	0.3161	20	0.8047

画图如下：

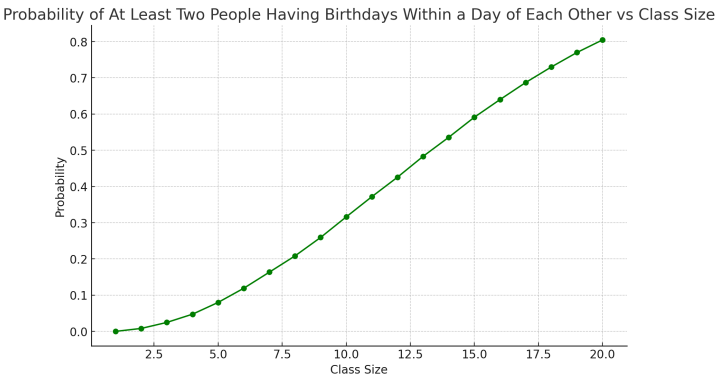


图 3　至少两个人的生日相差不过一天的频率

仿真结果和理论计算结果基本一致。

注：关于环形不相邻方案数的求解

这部分内容属于组合数学，但在本文求两人生日相邻问题时发挥至关重要的作用，故在此简单解释公式的计算方法。 要解环形不相邻问题，需先解链式不相邻问题。

链式不相邻问题

给定 n 颗小球，其中有 m 颗蓝色且其他状态相同小球，其余 $n-m$ 颗为红色相同的小球。求一种线性排列使得任意两个蓝色小球不直接接触的方案总数。

分析

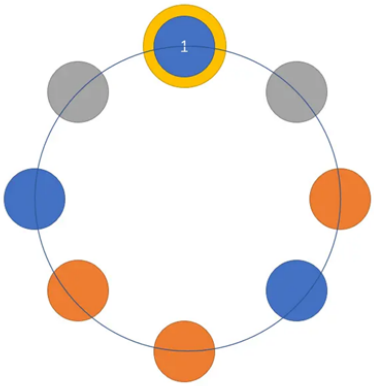
其中黄色区域为必须填入橘红色小球的区域，灰色区域为可填可不填区域。

换种理解角度，也就是说将 m 颗灰色小球变为蓝色，然后再取消掉剩余的灰色小球。这里一共有 $n-m+1$ 个灰色小球，所以答案为 C_{n-m-1}^{m}

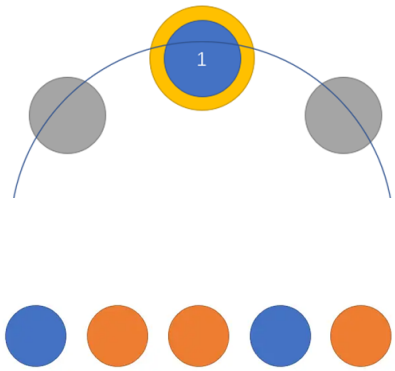
环形不相邻问题

有 n 个球排成一个环，从中取出 m 个不相邻的球，问有多少种解法。

分析



1. 先确定一个球必须选择为蓝色，则其两边的小球不能为蓝色。所以可以断开确定为蓝色的小球及其两边灰色小球的部分和其他剩余部分，将其他剩余部分看成是一个链。进而我们可以将其余部分的小球转化为第一类问题——链式不相邻问题。由于提前确定一个蓝球，于是剩下 $m-1$ 个蓝球，同时也剩下 $n-m-2$ 的橘红色的球。所以有 C_{n-m-1}^{m-1} 种方案



2. 如果这个球不是蓝色，则可将该球直接断开。所以此时转化成一个链式不相邻问题。一共有 m 个蓝球，$n-m-1$ 个橘红色的球的链式不相邻问题 于是有 C_{n-m}^{m} 种方案。

于是总共有 $C_{n-m}^{m}+C_{n-m-1}^{m-1}$ 种方案。

3 应用

生日问题的应用非常广泛，下面我们介绍几个常见的应用。

3.1 Pollard Rho 算法

Pollard Rho 算法是一种用于找出一个合数 N 的非平凡因子（即既不是 1 也不是 N 自身）的有效算法。这个算法使用一个随机选择的多项式函数 $f(x)$ 和两个初始点 x_1 和 x_2。然后，它迭代地应用这个函数，并检查是否在某一点找到了合适的因子。

Pollard Rho 算法是一种用于因数分解的算法，它利用了生日悖论的原理来寻找大数的因子。简单来说，这个算法通过随机选择和比较数的序列，试图找到某两个数的乘积模一个大数（即我们试图分解的数）结果相同的情况。这种情况下，根据生日悖论，我们意外地发现两个不同的数乘积的模有相同结果的概率比直观上看要高得多。

在 Pollard Rho 算法中，这意味着算法能够相对较快地找到大数的非平凡因子，尤其是当这个数不是一个大的素数时。这种方法在密码学中尤为重要，因为它可以用于攻击某些基于大数分解难度的加密系统，比如 RSA 加密中非常常用。

在 Pollard Rho 算法中，我们关心的是何时会有两个不同的 x 值（在模 N 下）满足 $f(x_1) \equiv f(x_2) \pmod{N}$。这样的话，$f(x_1) - f(x_2)$ 可能是 N 的一个非平凡因子。

这里的“碰撞” $f(x_1) \equiv f(x_2) \pmod{N}$ 类似于生日问题中至少两个人生日相同的情况。在生日问题中，我们知道，即使在相对小的人数 n 下，至少有两个人生日相同的概率也会很高。同样，在 Pollard's Rho 算法中，即使在较小的迭代次数下，也有很高的概率找到这样一个碰撞点。实际上根据生日问题可知序列中不同值的个数小于 $\sqrt{N}$。

生日悖论保证了 Pollard Rho 算法的期望时间复杂度为 $O\left(N^{\frac{1}{4}}\right)$，这比试除法的时间复杂度 $O(N)$ 要快得多。

3.2 哈希生日攻击

哈希生日攻击是一种利用生日悖论原理的密码学攻击方法。生日悖论指出，在相对较小的一组元素中，出现两个元素相同的概率意外地高。在哈希生日攻击的背景下，这意味着找到两个不同的输入，它们经过哈希函数处理后产生相同的哈希值（即哈希冲突），所需尝试的次数远少于直觉上的预期。

哈希函数通常被设计为将任意长度的输入转换为固定长度的输出（即哈希值）。在理想情况下，每个不同的输入都应映射到一个唯一的哈希值。然而，由于输出空间的有限性（例如，如果一个哈希函数产生 256 位的哈希值，那么总共只能有 2^{256} 个不同的哈希值），必然存在多个不同的输入对应于同一个输出的情况，这就是所谓的哈希冲突。

哈希生日攻击的基本思想是这样的：攻击者生成大量的随机输入，并计算它们的哈希值。根据生日悖论，攻击者只需要生成大约 $\sqrt{N}$ 个随机输入（其中 N 是哈希值的可能范围）就有很高的概率找到一对不同的输入，它们具有相同的哈希值。这种攻击对于寻找证书、数字签名或任何加密通信中的哈希冲突尤其有用。

例如，在一个理想的 256 位哈希函数中，攻击者可能需要尝试 2^{128} 次（而不是 2^{256} 次）来找到一对冲突。这种攻击对于理解和加强加密系统的安全性至关重要，它说明了为什么选择具有足够大输出空间的哈希函数对于保护信息安全非常重要。

为避免这种攻击，用于签名方案的哈希函数的输出长度应够大以从计算角度防止生日攻击。换言之，位数应为防止普通暴力破解所需位数的两倍。

3.2.1 原理

生日攻击通常用于针对哈希函数的攻击。哈希函数如 MD5 或 SHA 将无限的输入空间映射到有限的输出空间，因此不可能是单射映射。这意味着必然存在两个不同的输入 $M1$ 和 $M2$，它们的哈希值相同，即 $\mathrm{HASH}\ (M1) = \mathrm{HASH}\ (M2)$，这种情况称为碰撞。不同于解密过程，哈希破解不是为了还原原文，而是为了找到产生相同哈希值的不同输入。

在实践中，哈希函数被广泛用于校验数据的完整性。比如，在文件传输过程中，通过比对文件的 MD5 值可以验证文件是否被篡改。然而，如果有人能够产生一个恶意文件，其哈希值与官方文件的哈希值相同，这种强碰撞可能导致用户错误地信任并执行了恶意文件。

还有一种情况，比如某人声称能预测下一届世界杯的冠军，但只提供一个哈希值而不公开预测内容。世界杯结束后，此人发布一句声明如“中国赢得了世界杯”，并声称这句话的哈希值与之前提供的哈希值相匹配。这种手段的诡计在于，可以提前准备多个可能的结果的声明（如“中国赢得了世界杯”、“巴西必胜”、“德国获胜了”等），事实上根据哈希函数它们具有相同的哈希值，然后根据实际结果选择公布。因为每个词语若选择近义词，很容易生成指数级别不同的字符串，这些字符串中间很可能就会有互相碰撞的。

要找到一组碰撞，可以采用生成大量含义相似但不同的字符串的策略，比如通过变换近义词或短语。这就像询问走进一个教室需要多少人在场，以便有很大概率找到与你同一天生日的人。通过计算，我们可以得知，相比简单地随机选择，需要更多的个体来确保找到一个与给定日期相同的生日。这说明要找到哈希碰撞，需要的尝试次数可能比预期的要多。

3.2.2 理论计算

在密码学中，我们通常关注的是哈希函数 $f(x) = y$ 的性质，特别是当输入 $x_1 \neq x_2$ 时，哈希函数 f 返回的结果 y 相同的概率，这种情况被称为**哈希冲突**。找出一对碰撞的方法可以是，随机或伪随机地输入不同的数值，直到找出至少两个相同的结果为止。

对于一个哈希函数假如有 N 个等价关系（哈希值相同）类，我们将随机均匀地选取 n 个值（允许重复），使得至少一个等价类被选择至少一次。根据生日悖论可以通过以下公式来表示：

$$p(n, N) \approx 1 - \exp\left(\frac{-n(n-1)}{2N}\right) \approx 1 - \exp\left(\frac{-n^2}{2N}\right)$$

使 $n(p, H)$ 为我们将选择的最小数值，这种情况下找到碰撞的概率至少为 p。我们得到了下列估计公式：

$$n(p, N) \approx \sqrt{2N \ln \frac{1}{1-p}}$$

例如，当我们希望冲突概率至少为 50% 时，我们可以使用以下近似值：

$$n(0.5, N) \approx 1.1774\sqrt{N}$$

这表明，为了有较高的概率找到至少一个哈希冲突，我们需要进行的尝试次数与哈希空间的平方根成正比。举个例子，若使用 64 位哈希，则估计将有 1.8×10^{19} 个不同的输出。若这些输出均可

能发生（理想情况下），则攻击者“仅仅”需要约 50 亿次尝试 $\left(5.38 \times 10^9\right)$ 就能通过暴力攻击生成碰撞。这个结果通常被称为**生日界限** (birthday bound)。而对于 n 位密码则需要 $2^{\frac{n}{2}}$ 次尝试。

若函数的输出不平均分布，碰撞则可能将被更快找到。

更多举例如下，不同期望随机碰撞可能性对应了不同的破解次数：

位数	N	10^{-15}	10^{-12}	10^{-6}	0.1%	1%	25%	50%	75%
16	2^{16}	< 2	< 2	< 2	11	36	190	300	430
32	2^{32}	< 2	< 2	93	2900	9300	50000	77000	110000
64	2^{64}	190	6100	6100000	1.9×10^8	6.1×10^8	3.3×10^9	5.1×10^9	7.2×10^9
128	2^{128}	8.2×10^{11}	2.6×10^{13}	2.6×10^{16}	8.3×10^{17}	2.6×10^{18}	1.4×10^{19}	2.2×10^{19}	3.1×10^{19}
256	2^{256}	1.5×10^{31}	4.8×10^{32}	4.8×10^{35}	1.5×10^{37}	4.8×10^{37}	2.6×10^{38}	4.0×10^{38}	5.7×10^{38}
384	2^{384}	2.8×10^{50}	8.9×10^{51}	8.9×10^{54}	2.8×10^{56}	8.9×10^{56}	4.8×10^{57}	7.4×10^{57}	1.0×10^{58}
512	2^{512}	5.2×10^{69}	1.6×10^{71}	1.6×10^{74}	5.2×10^{75}	1.6×10^{76}	8.8×10^{76}	1.4×10^{77}	1.9×10^{77}

SATA 硬盘一个 bit 位出现数据错误的概率在 10^{-18} 到 10^{-15} 之间。作为比较，我们将哈希碰撞的概率控制在 10^{-15} 范围内已经相对来说非常安全了。

> 实际上，128 位的 MD5 哈希算法，输入数据保持在 8.2×10^{11} 个以内已经足够安全，基本不会产生哈希碰撞。SHA-256 的输入样本需要保持在 1.5×10^{31} 以内；SHA-512 的输入样本需要保持在 5.2×10^{69} 以内。

3.3 生日攻击解决离散对数问题

离散对数实际上是一个求解离散对数方程的问题，即求解 $a^x \equiv b \pmod p$ 中的 x。这个问题是一个 NP 问题，目前还没有找到多项式时间内求解的算法，大部分情况下其较难计算。在密码学中，基于这一点人们设计了许多非对称加密算法。

然而我们根据生日悖论，我们可以根据生日攻击来解决离散对数问题。当然也可以理解为取模就是一种哈希操作，利用哈希生日攻击来解决这个问题。

3.3.1 随机解法

生成两个长度为 $\sqrt{p}$ 的集合：

第一个集合包含 $a^A \bmod p$，通过随机选取 $\sqrt{p}$ 个 A 得到；

第二个集合包含 $ba^{-B} \bmod p$，通过随机选取 $\sqrt{p}$ 个 B 得到；

根据生日悖论，两个列表中很有可能出现重复的项， 即 $a^A = ba^{-B}$， 因此 $a^{A+B} = b \pmod p$。

$x = (A + B) \bmod (p - 1)$ 就是要找的离散对数 。

> **为什么是 $p - 1$?**
>
> 此处通过群论和数论的知识加以解释。

在离散对数问题中，当你在模 p 的环境下工作时（其中 p 是质数），对于任意基 a 和结果 b，你需要找到一个指数 x 使得 $a^x \equiv b \pmod{p}$。问题的关键在于，当 a 是模 p 下的一个生成元时，a 的所有可能的幂实际上形成了模 p 下的一个循环群，这个循环群的阶是 $p-1$。这意味着 $a^{p-1} \equiv 1 \pmod{p}$，而不是 $a^p \equiv 1 \pmod{p}$。

费马小定理说明了这一点：如果 p 是质数，而 a 是任何不被 p 整除的整数，则 $a^{p-1} \equiv 1 \pmod{p}$。

因此，所有大于 $p-1$ 的指数的幂都可以通过取模 $p-1$ 来简化。这就是为什么当我们找到等式 $a^A \equiv ba^{-B} \pmod{p}$ 时，我们实际上得到了 $a^{A+B} \equiv b \pmod{p}$，并且 $x = (A+B) \bmod (p-1)$ 是离散对数问题的解。简而言之，这是因为模 p 下的指数实际上是在模 $p-1$ 下循环的。

3.3.2 大步小步算法（Baby-Step Giant-Step, BSGS）

直接随机选取 A 和 B 的方法在时间复杂度的角度上并不可靠，而且一般并不能得到所有符合条件的 x，于是大步小步算法被提出。实际上这个算法是基于上面生日攻击朴素解法的改进，但内在思想还是生日悖论。

令 $x = At - B$，则有 $a^{At-B} \equiv b \pmod{p}$，稍加变换，则有 $a^{At} \equiv ba^B \pmod{p}$。

已知 a, b，我们可以先算出等式右边的 ba^B 的所有取值，枚举 B 存下来，再逐一计算 a^{At}，枚举 A，寻找是否有与之相等的 ba^B，从而可以得到所有的 x，$x = At - B$。

由欧拉定理 $a^{\varphi(p)} \equiv 1 \pmod{p}$，$B$ 的取值有 $\varphi(p) \bmod t$ 种，A 的取值有 $\left\lfloor \frac{\varphi(p)}{t} \right\rfloor$ 种，我们要让 $t + \left\lfloor \frac{\varphi(p)}{t} \right\rfloor$ 最小，根据均值不等式取 $t = \left\lceil \sqrt{\varphi(p)} \right\rceil$ 是最好的，当然我们认为 $\varphi(p)$ 和 p 同阶（因为若 p 为质数，$\varphi(p) = p-1$），为了避免计算欧拉函数，我们直接取 $t = \lceil \sqrt{p} \rceil$，此时取 $A, B \in [1, t]$ 可以保证每次 $x \in [1, p-1]$ 至少被覆盖一遍。时间复杂度为 $O(\sqrt{p})$。

为什么要求 a 与 p 互质?

注意到我们求出的是 A, B，我们需要保证从 $a^{A\lceil\sqrt{p}\rceil} \equiv ba^B \pmod{p}$ 可以推回 $a^{A\lceil\sqrt{p}\rceil - B} \equiv b \pmod{p}$，后式是前式左右两边除以 a^B 得到，所以必须有 $a^B \perp p$ 即 $a \perp p$。若 a 和 p 不互质，那么需要使用扩展 BSGS 算法，这里不再赘述。

4 结论

本文对生日问题及其在密码学中的应用进行了全面的研究。通过结合理论计算和基于 Python 的仿真模拟，我们还深入探讨了至少三人同日生日的概率以及至少两个人生日相差不超过一天的概率。我们的发现不仅验证了生日问题在概率论中的经典理解，而且还揭示了其在更大样本和更复杂场景中的行为模式。

在探讨生日问题在密码学中的应用时，特别是关于 Pollard Rho 算法和哈希生日攻击的分析，本文展示了生日问题在确保网络安全和加密协议中的关键作用。我们的理论分析和仿真结果为理解这些复杂算法提供了新的视角，并指出了它们在实际应用中的潜在局限性。

尽管我们的研究取得了一定的成果，但我们也认识到它的一些局限。例如，仿真模型的精确度受限于模拟的规模和假设的简化。未来的研究可以通过扩展仿真规模和采用更复杂的模型来克服这些局限。此外，考虑到生日问题在密码学中的重要性，探索新的算法和加密技术以抵御生日攻击是未来研究的另一个重要方向。

总体而言，本文对生日问题的理论和实践分析提供了拙见，为进一步探索概率论在现代科技中的应用提供了见解。