**西安电子科技大学网信院**

**信息安全基础与密码学**

**综合实验**

**实 验 报 告（一）**

**Fermat素性检测算法**

**班级：2118021**

**姓名：**

**学号：**

**日期：2023年10月22日**

一、实验目的

1. 实验环境

macOS Sonoma、Python 3.9

1. 实现目标

对于给定大整数，能够使用Fermat素性检验算法，判断其在给定概率下是否是素数，并编程实现Fermat大素数检验。

二、方案设计

1. 背景

在如RSA等现代密码学算法中，通常需要大素数才能实现加密算法。而遍历寻找大素数相对困难，因此常常给出大整数并进行素性检验，判断其是否为素数，从而选择出满足要求的大素数。

1. 原理

**1.费马小定理：**设为素数，则对于任意，有。即对于整数，。若，则为合数，否则为伪素数。

**2.快速模指数运算：**假设，，且将以二进制表示为，其中，。则模指数运

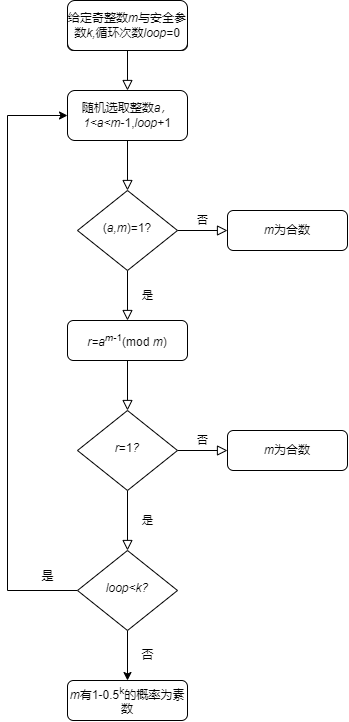
1. 步骤

给定要检验的整数与安全参数。

* 1. 随机选取整数；
  2. 计算。若，则转(3)；否则判断为合数，跳出循环。
  3. 计算，若，则判断为合数，跳出循环；若，且判断次数小于，则转(1)；否则有的概率为素数。

三、方案实现

1. 算法流程图



1. 主要函数的介绍
2. gcd(a, b)

计算的最大公因数

1. modular\_exponentiation(a, p, m)

使用快速模指数运算计算。

1. main()

接收给定奇整数与安全参数，并根据计算结果判断是否在给定概率下是素数。

1. 算法实现主要代码
2. import random

5. def gcd(a, b):
6. """
7. 用欧几里得算法计算最大公因数
8. """
9. if a < b:
10. temp = a
11. a = b
12. b = temp
14. if a % b == 0:
15. return b
17. return gcd(b, a % b)

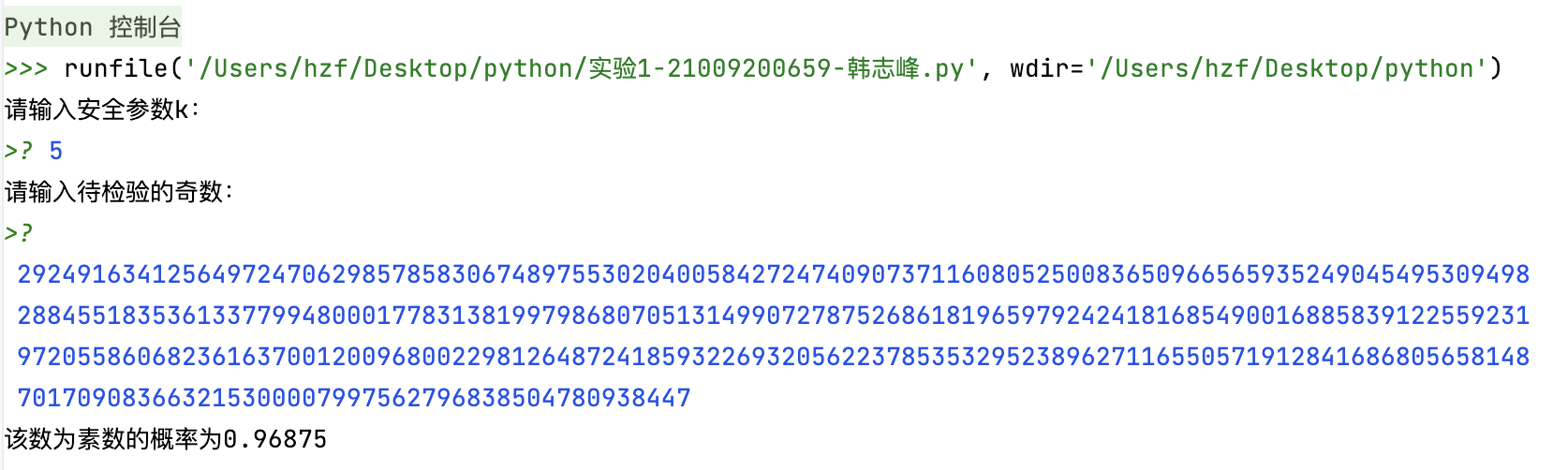
20. def modularexponentiation(a, p, m):
21. """
22. 快速模指数运算，a^p(mod m)
23. """
24. binaryp = bin(p)
25. reversedbinaryp = binaryp[len(binaryp):1:-1]
27. # 运算结果
28. result = 1
29. # 每一个二进制位对应底数
30. bn = a
31. for n in range(len(reversedbinaryp)):
32. result = result bn int(reversedbinaryp[n]) % m
33. bn = bn 2 % m
35. return result

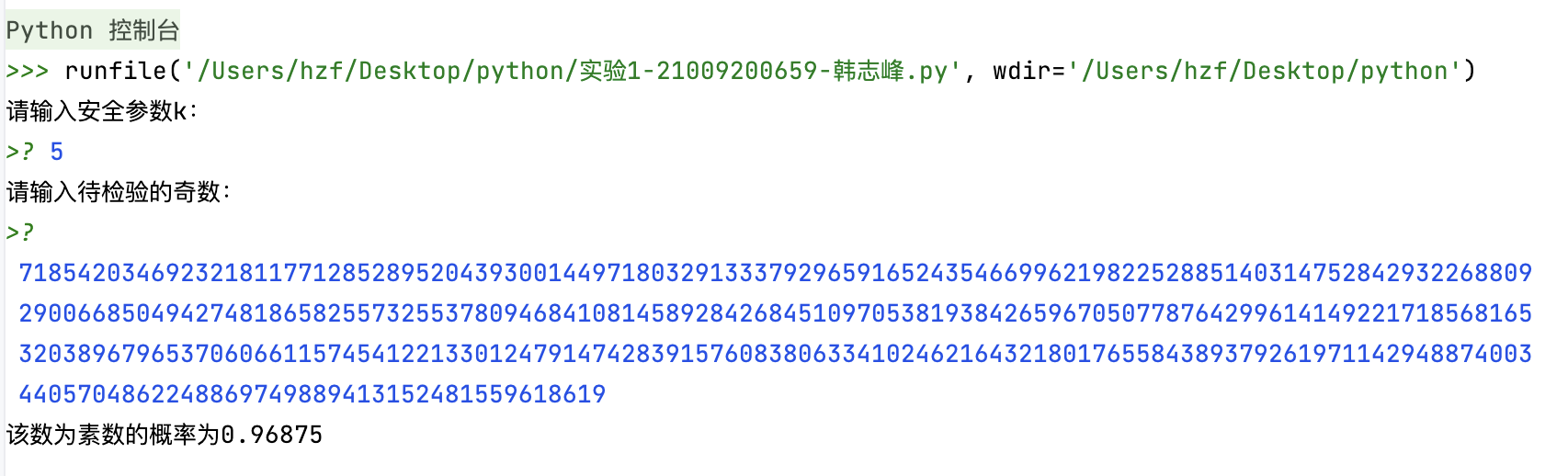
38. def main():
39. k = 0
40. print("请输入安全参数k：")
41. k = int(input())
43. m = 0
44. print("请输入待检验的奇数：")
45. m = int(input())
47. if m % 2 == 0:
48. print("该数为合数")
49. return 0
50. if m < 3:
51. print("请输入不小于3的奇数")
52. return 0
53. if m == 3:
54. print("该数为素数")
55. return 0
57. for i in range(k):
58. a = random.randint(2, m - 2)
59. if gcd(a, m) != 1:
60. print("该数为合数")
61. return 0
62. if modularexponentiation(a, m - 1, m) != 1:
63. print("该数为合数")
64. return 0
66. print("该数为素数的概率为", 1 - 0.5 k, sep='')

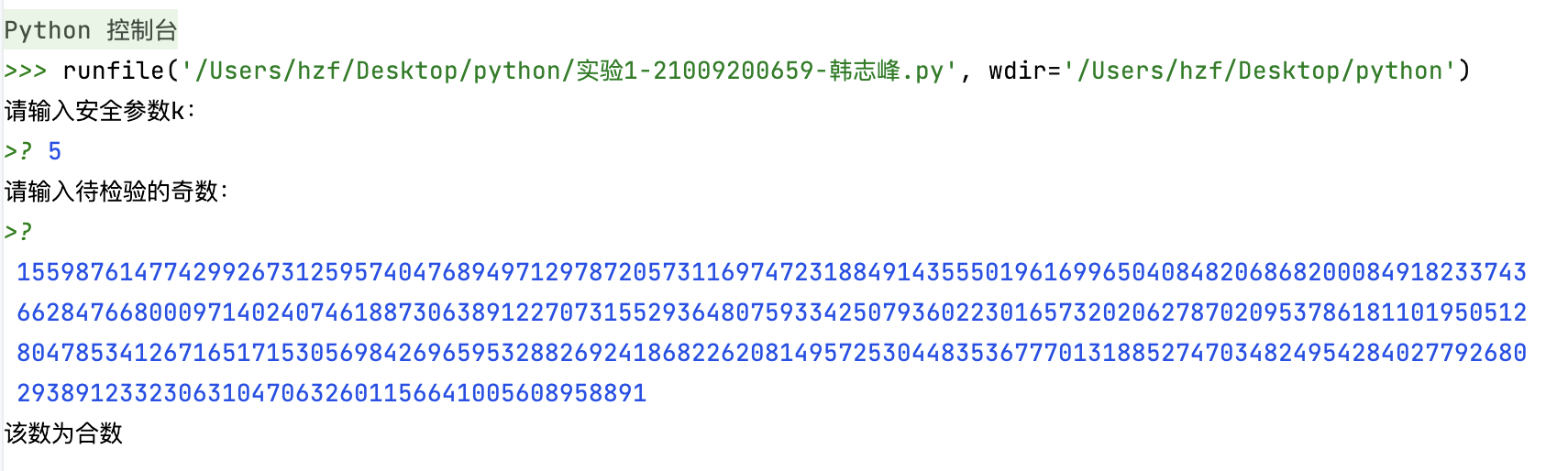
69. if name == 'main':
70. main()

四、数据分析

选定安全参数*k*=5，使用测试数据1、2、3、4进行测试。运行结果如下：









五、思考与总结

1. 如果有一个整数𝒂，(𝒂,𝒎)=𝟏，使得𝒂𝒎−𝟏≡𝟏 𝒎𝒐𝒅 𝒎 则𝒎一定是一个素数吗？为什么？（请简述并举例说明，不能只简单回答“是”或“不是”）

不是。根据费马小定理，对于整数，且，仅为m是素数的必要条件。Fermat小定理仅能确定性证明一个数是合数，而仅在一定概率下证明一个数是素数。

例如，当，时，易得，而是合数，即不能通过上述结论判断一定是一个素数。

1. Fermat素性检测中都用到了哪些运算？分别实现什么功能？请简述。

使用运算如下：

* 1. 生成随机数

用于随机选取整数。

* 1. 计算最大公因数——欧几里得算法

用于在步骤2中计算的最大公因数，判断是否是素数。

* 1. 快速模指数运算

用于在步骤3中计算。

1. 你还了解哪种素性检测算法？请简述，并分析其与Fermat素性检测算法的区别与联系。

Eratosthenes筛法。

由定理可知，对于不小于2的整数，若对于任意素数，都不是的因数，则为素数。因此通过判断不大于的素数中，是否有的因数，即可判断是否为素数。若存在的因数，则为合数，否则为素数。

区别：Fermat素性检验算法是概率算法，而Eratosthenes筛法是确定性算法；在对大整数进行素性检验时，Fermat能够较快地在一定概率下判断出结果，而Eratosthenes筛法对此运行速度较慢。

联系：两种算法均是根据已有定理而设计的素性检验算法。

1. 实验过程中还遇到了什么问题，如何解决的？通过该实验有何收获？

遇到的问题：

素性检验计算时间太长。应使用快速模指数算法计算，且在计算时，应将结果及时通过同余运算缩小结果，避免指数爆炸导致计算时间过长。

收获：

通过本次实验，我对Fremat素性检验算法有了更深地理解，同时对快速模指数运算的快速性有了更深的认识，对于数学基础的相关知识也有了进一步的巩固与加强。