**Fermat素性检测算法**

一、实验目的（包括实验环境、实现目标等等）

1.实验环境：python 3.11.3

2.实验目标：理解 Fermat 素性检测算法的基本原理和费马小定理，并且编写相应程序实现对大数实现素数检验。

二、方案设计

（包括背景、原理、必要的公式、图表、算法步骤等等）

背景：素性检测是判断一个数是否为素数的过程。素数在密码学、数论等领域有着重要的应用。本次实验旨在运用Fermat 素性检测来检验4个大数是否为素数。

原理：费马小定理指出若 p 为素数，并且 a 是一个小于 p 的整数，满足。基于此定理得到Fermat素性检测。

给定奇整数m≥3和安全参数k

（1）随机选取整数a,2≤a≤m-2；

（2）计算，如果，转(3);否则，跳出，m为合数；

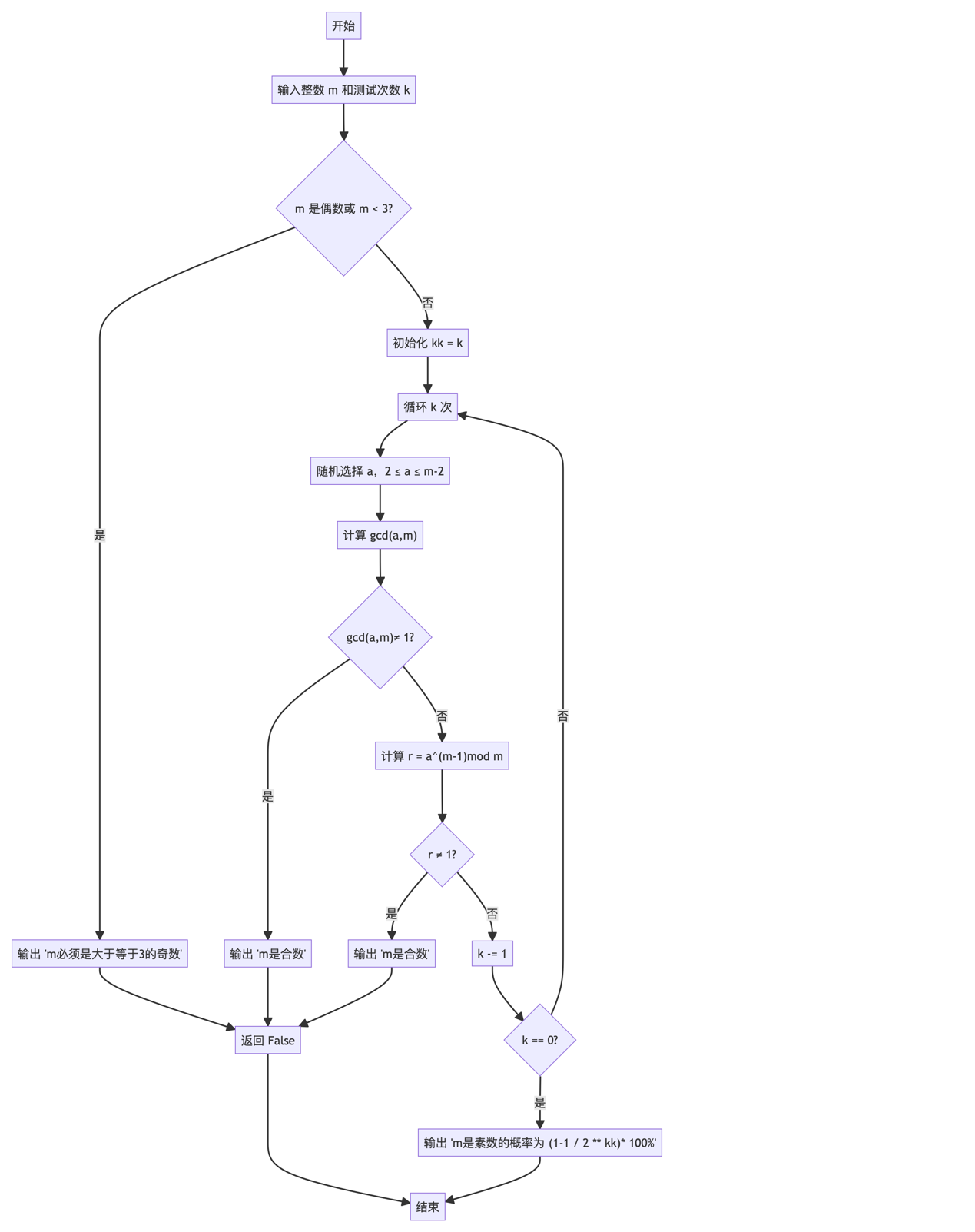
（3）计算，如果r=1，m可能是素数，转（1）；否则，跳出，m为合数；

（4）重复上述过程k次，如果每次得到m可能为素数，则m为素数的概率为。

三、方案实现

（包括算法流程图、主要函数的介绍、算法实现的主要代码等等）

流程图：



主要函数介绍：

1. 计算最大公约数的函数gcd(a,b):

 def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

这个运用迭代的思想，运用欧几里得算法不断地将每轮的余数和除数互换后相除更新余数的思想得到最终的最大公约数。

2．快速模指数算法exponentiation(a, b, n)

def exponentiation(a, b, n):

c = 1

d = a % n

while b:

if b % 2 == 1:

c = c \* d % n

d = d \* d % n

b //= 2

return c

在本函数中运用快速模指数算法，这个是用来计算的其中c用来存储每个阶段的中间变量，d用来存储每轮的数值。这样能够快速得到最终答案。

1. 用fermat(m,k)函数来检验在安全系数为k的情况下m是否为素数

def fermat(m,k):

if m % 2 == 0 or m < 3:

print("m必须是大于等于3的奇数")

return False

kk = k

while k:

a = random.randint(2, m - 2)

g = gcd(a, m)

if g != 1:

print(f"m是合数")

break

else:

r = exponentiation(a, m - 1, m)

if r != 1:

print(f"m是合数")

break

k -= 1

if k == 0:

print(f"m是素数的概率为{(1 - 1 / 2 \*\* kk) \* 100}%")

主要思路是首先判断m是否为大于等于3的奇数，然后判断m和生成的随机数a是否互素，然后再计算r是否数值为1以此判断是否是合数，重复k论实验，若k论实验成功最终得到m为素数概率为，反之m为合数。

1. 最后调用函数检验

主函数如下

if \_\_name\_\_ == "\_\_main\_\_":

f = open("4.txt", "r")

m = int(f.readline())

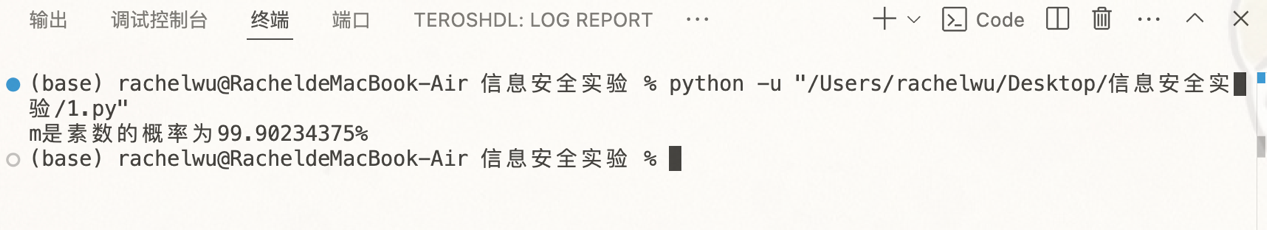
f.close()

k = 10

fermat(m, k)

四、数据分析(包括算法测试数据的分析，运行结果截图等等)

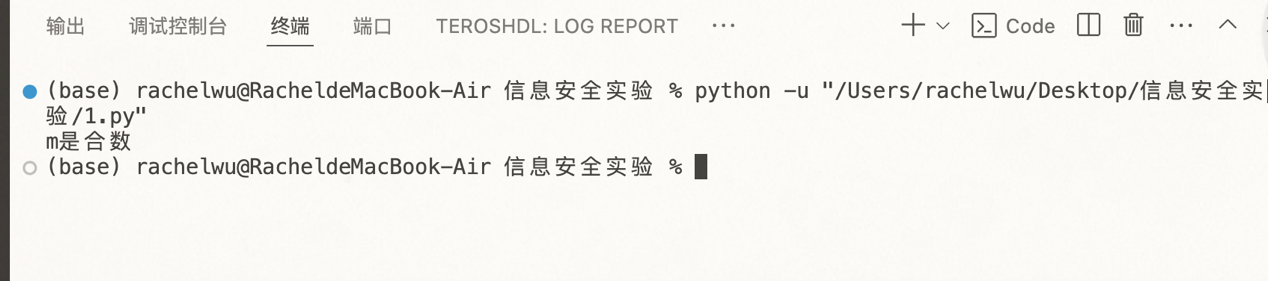
1. 对1.txt检验：k = 10



1. 对2.txt检验：k = 10



1. 对3.txt检验：k = 10



1. 对4.txt检验：k = 10



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”）

答：m不一定是素数，因为费马小定理是规定m 是一个素数，那么对于任何整数 a（），都有 ，但反过来并不一定成立。存在一些合数 m 也满足上述条件，是伪素数。

比如：我们取m = 341, a = 2,很明显,但是

又因为可以说明m是合数。

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。

答：运用了用欧几里得算法求最大公约数，用来检验随机生成的a是否和m互素。运用了快速模指数算法运算，用来快速计算r = m-1≡1 mod m的数值，判断m是否为合数。

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。

答： Miller-Rabin 素性检测算法，它是一种基于随机性的概率素性检测算法。也是多次测试来判断一个数是否为素数。

算法：

1. 检查m是否为 2 或 3，如果是，则 n 是素数。检查 n 是否为偶数或小于 2，如果是，则 n 是合数。

2. 将待检测的数 m 写成的形式，d为奇数。

3. 进行 k 次测试：

(1)随机选择一个整数 a，(。

(2)计算 。如果或，则 m 可能是素数。则继续下一次测试转到(1);否则重复计算，最多重复次。如果在任何一次计算中，则继续下一次测试转到(1)。如果所有测试都未通过，则m是合数。

(3)如果 k 次测试都通过，则m可能是素数,m为素数概率为

区别：

Fermat 素性检测只进行一次幂模运算，而 Miller-Rabin 进行多次幂模运算，检测更严格。Fermat 素性检测容易被费马伪素数欺骗，而 Miller-Rabin 通过多次检测减少了伪素数的概率。并且在安全参数为k的情况下Fermat 素性检测正确概率为，而Miller-Rabin正确概率是

联系：

两者都是随机性的概率素性检测的算法，并且都使用模指数运算来检验是否是素数。

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？

答：在快速模指数函数编写的时候起初运行的很慢，后来我发现是我在每轮生成中间变量的时候没有用取余数之后的值进行平方运算。后来改正后速度得到了提高。经过本实验我深入理解了 Fermat 素性检测算法的原理和实现过程。学会了如何使用快速幂算法进行高效的大整数运算。此外还了解了Miller-Rabin 素性检测算法。