**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：**

**姓名：**

**学号：**

**日期：2022.10.30**

一、实验目的（包括实验环境、实现目标等等）

采用Python实验环境。（Python3.10）

实现Fermat素性检测算法，给定一个大整数，利用Fermat素性检测算法是否为素数。

即给定奇整数m⼤于等于3和安全参数k，判断m是否为合数，或者有(1-1/2^k)概率为素数。

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，即检查一个数是否为质数的问题，而不需要对其进行质因数分解。因当今计算机的广泛使用，加密和伪随机数算法到处可见。而这些算法中，有许多都需要先产生一个随机大质数，所以素性检验问题对这些算法是至关紧要。

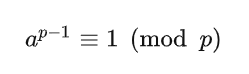
* 原理：

费马小定理的引入：

假如 a是一个整数， p 是一个质数，那么



也可写为：



那么p是否是素数，我们在选取a（2<a<p-1），看等式是否成立。如果对于数值a等式不成立，那么n是合数。如果存在a能够使等式成立，那么p可能是素数，或者[伪素数](https://baike.baidu.com/item/%E4%BC%AA%E7%B4%A0%E6%95%B0/9262664?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E8%B4%B9%E9%A9%AC%E7%B4%A0%E6%80%A7%E6%A3%80%E9%AA%8C/_blank)。各占1/2的概率。

所以通过此即可判断p是否为素数。

* 算法

给定奇整数m≥3和安全参数k=5

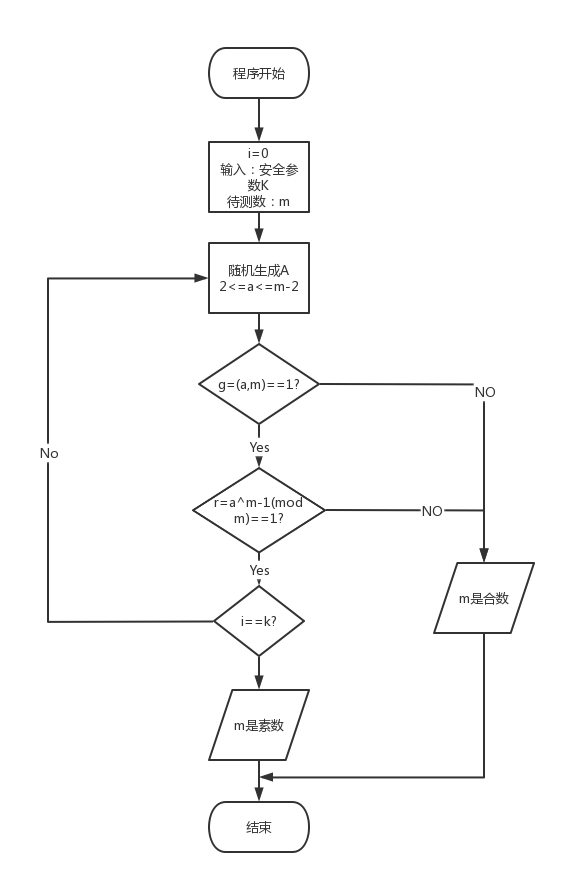
（1） 随机选取整数a,2≤a≤m-2

（2） 计算g=(a, m)，如果g =1，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。

流程图为：



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）

主要函数为

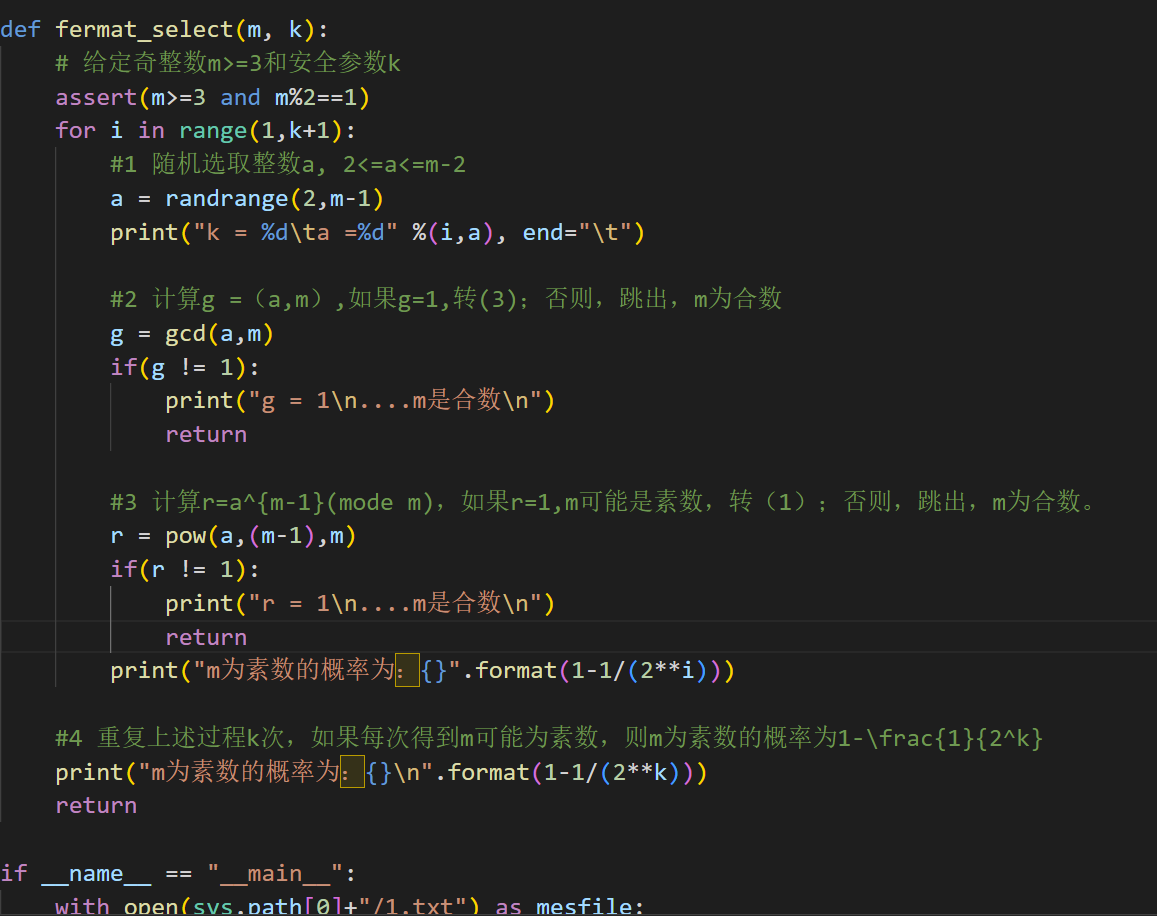
fermat\_select(m, k) //主要判断函数

a = randrange(2,m-1) //生成随机数a

gcd(a,m) //判断是否互素

pow(a,(m-1),m) //

主函数为：



先设定m的输入范围

取随机数a

计算g=（a,m）判断是否互素

不互素则为合数

若互素计算

如不等于1则为合数

如果为1则有的概率为素数，和循环次数有关。

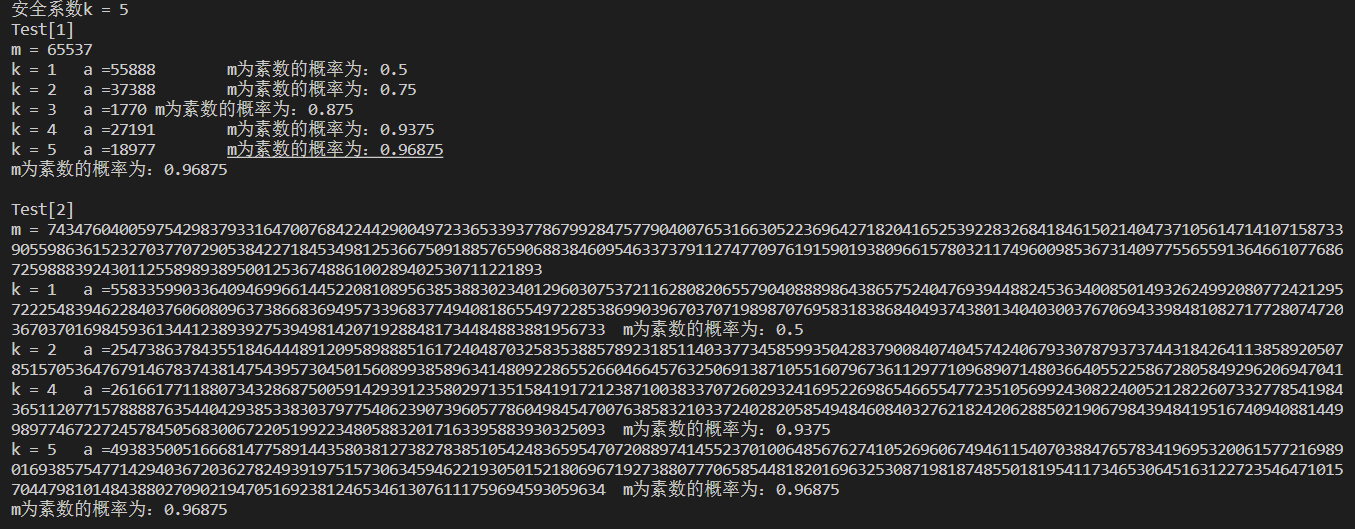
1. 数据分析(包括算法测试数据的分析，运行结果截图等等)

设置安全系数为5,则测试五次

选择奇数，和待测试大整数

观察结果

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”）

p可能是素数，或者[伪素数](https://baike.baidu.com/item/%E4%BC%AA%E7%B4%A0%E6%95%B0/9262664?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E8%B4%B9%E9%A9%AC%E7%B4%A0%E6%80%A7%E6%A3%80%E9%AA%8C/_blank)。各占1/2的概率

比如m为341

但m为合数 =11×13

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。

* 最大公因数运算，判断是否和a互素。若不互素肯定为合数。
* 同余运算，用于计算费马定理，判断是否满足费马小定理，满足则有1/2的概率是素数。

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。

米勒—拉宾测试算法：

它是基于广义黎曼猜想的确定性算法。也就是：如果广义黎曼假设被证明是真命题，那么这个算法的结果就是正确的，且是决定性的。但是黎曼假设的证明还遥遥无期，所以后来以色列的科学家Michael Rabin将其改进，使其不再依赖黎曼假设，但结果是概率性的。

简述为：

如果如果我们能找到这样一个a，使得对任意以下两个式子均满足：

那么n就不是一个素数。这样的a称为n是合数的一个凭证（witness）。否则a可能是是一个证明n是素数的“强伪证”（strong liar），即当n确实是一个合数，但是对当前选取的a来说上述两个式子均不满足，这时我们认为n是基于a的大概率素数。

区别与联系：

费马素性测试和米勒—拉宾测试算法都是判断一个数是否为素数的一个基于概率的测试。

事实上，费马小定理的逆否定理成立，而费马小定理的逆定理是不成立的，而费马素性测试就是基于费马小定理的“逆定理”的。所以不一定成立。

但是在速度上，米勒-拉宾测试有了质的飞跃。米勒-拉宾素性测试是当前运用最广泛的素性测试，且加上限制条件完全可以作为确定性算法。

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？

（1）算法要求随机数2≤a≤m-2，所以产生随机数a的时候，要引入Crypto.Random.random模块，要使用pip install pycryptodome。但是，在使用的时候导入模块是有问题的，这个时候只要修改一个文件夹的名称

Python\Python36\Lib\site-packages，找到这个路径，下面有一个文件夹叫做crypto,将小写c改成大写C就ok了。

（2）大数的比较

需使用compare()函数比较大小，不能直接用a==b。

本次实验学习了费马素性检测算法，解决了Python大整数编程问题，学习了密码学常使用的函数，最终实现了算法。