数学实验课程第二次实验报告

PB21010479　王曹励文

2023 年 9 月 19 日

1　求解 500 以内的素数

1.1　实验题目

通过如下方法编程求解 500 以内的所有素数：

1. 利用 Eratosthenes 筛法计算；
2. 利用试除方法计算；
3. 利用维路判别法计算；
4. 利用概率判别法计算；

并统计上述四种方法的计算时间，判断哪一个更有效.

1.2　解决思路

首先对于计算时间，我利用了 matlab 中内置的 tic、toc 函数，其使用规则如下：

内置的 tic、toc 函数

```
tic
% 放置代码
elapsed_time = toc; % 停止计时并获取经过的时间
disp(['执行时间：', num2str(elapsed_time), '秒']);
```

通过打印的计算时间即可以判断哪一个更加高效.

对于每一种方法，我均定义了 N 代表数的范围，布尔数组（即只有 true 与 false 的数组）is_prime 来判断是否为素数，值为 true 表示为素数，值为 false 表示不为素数；primes 数组来存储判断为素数的元素；最后利用对 n 的循环逐个打印素数，并打印计算时间.

1.2.1　Eratosthenes 筛法

对于 Eratosthenes 筛法，我首先将所有的数字均定义为 true，对于 p 从 2 开始，首先将其添加到 primes 列表中，再利用 for 循环，从 2 开始，步长为 2，即将 2 的倍数均给予 false 的值，等价于将其判

断为合数；再对下一个暂时判定为 true 的数进行类似的操作，从 p 开始，步长为 p，即将 p 的倍数均给予 false 的值，等价于将其判断为合数；执行多个循环，最终筛选出了 500 以内的素数. 实验结果如图 1 所示，源代码见文件 5.1.

1.2.2 试除方法

对于试除方法，我首先给予 primes 数组有一个初值为 2，代表第一个素数，然后对 n 从 3 到 N=500 循环，判断每一个 n 是不是素数，具体的判断方法为，对此时 primes 数组中的每一个数，判断 modn 的结果，若为 0，则 n 为合数，赋值 false，并跳出循环；若不为 0，则 n 为素数，赋值仍然为 true. 最后在对 n 的循环中，在 primes 中写入赋值为 true 的数，筛选出了 500 以内的素数. 实验结果如图 2 所示，源代码见文件 5.1.

1.2.3 维路判别法

对于维路判别法，由于内置函数 jacobiSymbol(a, n) 中 n 只能为奇数，因此先加入了对偶数的判断，>2 的偶数显然为合数，对他们赋值 false；对 n 为奇数，对 a 进行循环，由于需要对 n 取 mod，只需要考虑 a 从 2 到 min(n-1, ceil(10 * log(n)^2))，其中 ceil 表示上取整，目的为保持 a 为整数. 由于涉及指数上较大的运算，使用了 sym 进行求解. 根据判别法，如果存在定义中 a_mod 与 an_mod 相等的情形，则 a 被判定为合数，赋值 false 并跳出循环；最后对 n 循环，将 true 的值写入 primes 中，筛选出了 500 以内的素数. 实验结果如图 3 所示，源代码见文件 5.1.

1.2.4 概率判别法

对于概率判别法，先定义选取随机数次数 $k = 10$，并由于之后选取随机数范围时涉及到 [2,n-1]，因此将 2、3 素数写入 primes，从 n=4 开始循环. 对每个 n，循环 k 次，每次从 [2,n-1] 中选取一个随机数 p，如果 n 为 p 的倍数，则 n 为合数，赋值 false；如果 gcd(n,p)=1，进一步考虑 p^{n-1} 对 n 的余数 J，J 不为 1 或者-1 时，n 为合数，赋值 false；若 J 为 1 或者-1，p 不是素数的可能性最多是 50%，因此重复实验 10 次，该可能性极低. 过程中涉及了较大次幂运算，因此使用 sym 函数. 最后对 n 循环，将 true 的值写入 primes 中，筛选出了 500 以内的素数. 实验结果如图 4 所示，源代码见文件 5.1.

1.3 实验结果

从图 1 至图 4 可以看出，Eratosthenes 筛法具有最高的效率，而维路判别法的运行时间高达 40s，我认为这与其进行了较大规模的次幂运算有关；Eratosthenes 筛法通常是查找小于某个上限 N 的素数最有效的方法，因为它具有较低的时间复杂度.

```
列 1 至 17

  2    3    5    7   11   13   17   19   23   29   31   37   41   43   47   53   59

列 18 至 34

 61   67   71   73   79   83   89   97  101  103  107  109  113  127  131  137  139

列 35 至 51

149  151  157  163  167  173  179  181  191  193  197  199  211  223  227  229  233

列 52 至 68

239  241  251  257  263  269  271  277  281  283  293  307  311  313  317  331  337

列 69 至 85

347  349  353  359  367  373  379  383  389  397  401  409  419  421  431  433  439

列 86 至 95

443  449  457  461  463  467  479  487  491  499

执行时间：0.028895 秒
```

图 1: Eratosthenes 筛法实验结果

```
列 1 至 17

  2    3    5    7   11   13   17   19   23   29   31   37   41   43   47   53   59

列 18 至 34

 61   67   71   73   79   83   89   97  101  103  107  109  113  127  131  137  139

列 35 至 51

149  151  157  163  167  173  179  181  191  193  197  199  211  223  227  229  233

列 52 至 68

239  241  251  257  263  269  271  277  281  283  293  307  311  313  317  331  337

列 69 至 85

347  349  353  359  367  373  379  383  389  397  401  409  419  421  431  433  439

列 86 至 95

443  449  457  461  463  467  479  487  491  499

执行时间：0.038974 秒
```

图 2: 试除方法实验结果

2 素数的分布

2.1 实验题目

将素数在数轴上标出来，观察素数的分布，试图通过以下实验进行进一步的观察：

用 $\pi(n)$ 表示不超过 n 的素数的个数, $\pi(m,n)$ 表示区间 $[m,n]$ 内素数的个数, 试计算 $\pi(100)$、$\pi(1000)$、$\pi(10000)$ 以及 $\pi(100,200)$、$\pi(1000,1100)$、$\pi(10000,10100)$. 并从计算结果上看，随着整数范围的扩大，素数是越来越稀还是越来越密.

```
列 1 至 17

    2     3     5     7    11    13    17    19    23    29    31    37    41    43    47    53    59

列 18 至 34

   61    67    71    73    79    83    89    97   101   103   107   109   113   127   131   137   139

列 35 至 51

  149   151   157   163   167   173   179   181   191   193   197   199   211   223   227   229   233

列 52 至 68

  239   241   251   257   263   269   271   277   281   283   293   307   311   313   317   331   337

列 69 至 85

  347   349   353   359   367   373   379   383   389   397   401   409   419   421   431   433   439

列 86 至 95

  443   449   457   461   463   467   479   487   491   499

执行时间：41.6505 秒
```

图 3: 维路判别法实验结果

```
列 1 至 17

    2     3     5     7    11    13    17    19    23    29    31    37    41    43    47    53    59

列 18 至 34

   61    67    71    73    79    83    89    97   101   103   107   109   113   127   131   137   139

列 35 至 51

  149   151   157   163   167   173   179   181   191   193   197   199   211   223   227   229   233

列 52 至 68

  239   241   251   257   263   269   271   277   281   283   293   307   311   313   317   331   337

列 69 至 85

  347   349   353   359   367   373   379   383   389   397   401   409   419   421   431   433   439

列 86 至 95

  443   449   457   461   463   467   479   487   491   499

执行时间：1.6844 秒
```

图 4: 概率判别法实验结果

2.2 解决思路

从上一题的结论中我们知道 Eratosthenes 筛法具有最高的效率，因此我们直接使用 Eratosthenes 筛法计算本题，由于要计算多个 N 对应的值，加入循环结构即可，实验结果如图 5 所示，源代码见文件 5.2.

对于 $\pi(n)$ 的计算我们直接将 n 的值赋予 N；对于 $\pi(m,n)$ 的计算，根据定义

$$\pi(m,n) = \pi(n) - \pi(m-1).$$

注意到由于 n>m，我们只需要计算不超过 n 的素数，写入 primes 数组后，去除 <m 的素数，再对后来

的 prime 求长度即可知道 $\pi(m,n)$ 的值. 实验结果如图 6 所示，源代码见文件 5.2，从图 6 的计算结果可以看出素数是越来越稀疏的.

最后再将相应区间内的素数标记为红色，作出可视化图像，观察素数的分布情况. 具体操作为先画出数轴，再将 primes 中的点标记为红色，实验结果如图 7、8、9 所示，源代码见文件 5.2.

2.3 实验结果

```
π(100)的值为：25
π(1000)的值为：168
π(10000)的值为：1229
```

图 5: $\pi(n)$ 计算结果

```
当 M = 100 时，π(200) - π(99) 的结果为：21
当 M = 1000 时，π(1100) - π(999) 的结果为：16
当 M = 10000 时，π(10100) - π(9999) 的结果为：11
```

图 6: $\pi(m,n)$ 计算结果

图 7: $\pi(100,200)$ 可视化结果

图 8: $\pi(1000,1100)$ 可视化结果

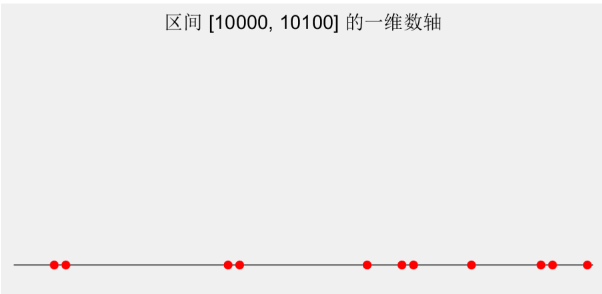

图 9: $\pi(10000, 10100)$ 可视化结果