

文章编号:1005-0523(2003)02-0099-03

Eratostheness 筛法的推广与素数分布

周尚超, 刘二根

(华东交通大学 基础科学学院, 江西 南昌 330013)

摘要:将计算机素数的筛法进行改进并将其推广为分段筛法, 扩大计算范围, 提高了运行速度, 计算出 100 亿以下素数表和 2 000 亿以下的素数分布.

关键词:素数; 筛法; Delphi

中图分类号:0157.5

文献标识码:A

0 引言

文献^[1]列出了小于等于 10 亿以下的素数表及小于 1 百万的素数对的分布, 本文通过将计算素数的筛法进行改进并将其推广为分段筛法, 扩大了计算范围, 提高了运行速度, 计算出 2 000 亿以下素数表和素数分布. 另外, 在组合论与数论的一些数值计算中, 经常要用到素数的计算. 例如, 在我们的某个计算项目中, 要将许多小于 10^{20} 的数分解为素因子之积, 这就经常要用小于 100 亿的素数. 小于 100 亿的素数有约 5 亿个, 我们将这些素数分为 1 000 段, 每段为连续的 1 000 万个数中素数, 每段的素数存放在 1 个文本文件中, 每个文件约占 5M 的磁盘空间, 1 000 个文件共占 5G 的磁盘空间. 这在 10 年以前, 用微机进行这种计算几乎是不可能的, 那时的软件如 BorlandC++, 数组不能大于 64k, 定义 1 个存放素数的数组只能存入约 32 000 个, 而且那时微机的硬盘也不到 1G. 现在我们用内存 128M 的 P4-1.8G 的微机计算, 可定义 1 个存放 (15 段) 150 000 000 以下的所有素数的数组, 再定义 1 个存放 1 段素数的临时数组, 当前 15 段数组不够用时, 读入第 16 段于临时数组, 在不够用时又读入第 17 段于临时数组, 一直到最后 1 段.

目前有的书上介绍的素数的计算都是为了教学, 没有考虑到科学计算. 例如判断素数 m ; 如 m 不能被小于 m 的平方根的所有奇数整除, 则 m 是素数. 我们把它改进为: 如 m 不能被小于 m 的平方根的所有素数整除, 则 m 是素数. 素数比奇数少得多, 这就减少了许多计算. 因此我们编制的算法算出了 42 亿以下的素数 (分为 420 段计算, 第 1 段是 10 000 000 以下的素数, 第 2 段是 10 000 001 ~ 20 000 000 间的素数等, 这样计算 1 段只要 30 秒). 另外, 在目前的计算软件 VC++ 6 中, 整数表示的最大值约为 21 亿. 而用 Delphi 计算, 例 Delphi 4 以上版本无符号整数变量 (cardinal) 最大值约为 42 亿 ($2^{32} - 1$). 在超过 42 亿时, 用 8 字节整数 (int64), 它可表示的最大整数为 922 亿亿 ($2^{63} - 1$), 这时计算 1 段要 1 小时.

由于用的是除法, 对每个数要用多次除法, 计算速度慢, 但它计算范围大, 例如它可 1 次计算出 42 亿以下所有素数. 筛法用的是加法和乘法, 计算速度快, 但占用内存空间较大, 它用数组 Isprime[i] = 1 表示 i 为素数, Isprime[i] = 0 表示 i 不是素数. 在 128M 内存的计算机上数组 Isprime 最大可定义到 10 000 000, 可计算 10 000 000 以下的素数. 因此目前的筛法的计算范围受到计算机内存的限制. 计算时, 首先置所有 Isprime[i] = 1, 然后将 2 的所有倍数置 0, 3 的所有倍数置 0, 5 的所有倍数置 0, ..., 就象

收稿日期: 2002-12-10

作者简介: 周尚超 (1948-), 男, 云南蒙自人, 教授.

筛子把它们筛掉. 以下算法就是筛法, 它计算小于等于 m 的所有素数, 当 $m = 10\ 000\ 000$ 时计算时间为 1 秒.

1 算 法

算法 1(筛法):

```

k := 1;
for i := 1 to do if odd(i) then Isprime[i] := 1
else Isprime[i] := 0; //先删除所有偶数
for i := 3 to m do begin
    if Isprime[i] := 1 then begin
        n := i + i;
        while n <= m do begin
            Isprime[n] := 0;
            n := n + i;
        end;
    end;
end;
```

算法 1 是根据公元前约 250 年, 古希腊数学家 Eratostheness 提出的计算不超过 m 的素数的方法(即筛法)编制的程序. 1914 年 D. N. Lehmer 发表了 1 到 10 006 721 的素数表. 1951 年 Kulik 又把它增加到 10 999 997.

我们将计算从 1 到 m 的素数的筛法推广到计算从 m_1 到 m_2 的素数的分段筛法并对原筛法作了改进, 同时此改进也用于分段筛法. 计算出的 2 000 亿下的以素数存于 1 000 个文本文件. 每个文件平均 50000 行, 这是无法在刊物上刊登的, 可通过向下列地址发电子邮件索取: zhousc@ecjtu.jx.cn

算法 2(分段筛法):

计算 m_1 到 m_2 的所有素数, m_1 为奇数, $m_2 \sim m_1 < = 10\ 000\ 000$, 并存储在文件 primeM2.txt 中. 设小于 m_2 平方根的 $pNum$ 个素数 $primes[1] := 2, primes[2] := 3, \dots, primes[pNum]$ 已算出 $isprime[i] := 1$ 表示 $m_1 + i - 1$ 是素数, $isprime[i] := 0$ 不是素数. 函数 prime 的返回值是 m_1 到 m_2 的素数的个数.

```

function prime(m1, m2: int64): integer;
var
    i, j, k, s, t: integer; st: string;
    mp, ms: int64;
    isprime: array[1..10 000 000] of byte;
```

```

begin
    st := 'primeM2.txt';
    assignfile(outfile, st);
    rewrite(outfile);
    mp := m1 - 1;
    for i := 1 to 10 000 000 do
        if odd(i) then isprime[i] := 1
        else isprime[i] := 0;

    for i := 2 to pNum do begin
        s := primes[i];
        t := m1 div s;
        if m1 mod s < > 0 then inc(t);
        if not odd(t) then t := t + 1;
        //t * s 是恰好大于 m1 的奇数, 它是素数 s 的倍数
        //将 s 的倍数筛除: 设置 isprime[] := 0;
        n := m2 div s;
        ms := t * s - mp;
        for k := t to n do begin
            isprime[ms] := 0;
            ms := ms + s;
        end;
    end;

    //将素数存入文件, 每行 10 个
    k := 0;
    for i := 1 to 10 000 000 do begin
        if isprime[i] := 1 then begin
            ms := i + mp; k := k + 1;
            write(outfile, ms, ' ');
            if k mod 10 = 0 then writeln(outfile);
        end;
    end;
    closefile(outfile);
    prime := k;
end;
```

2 计算结果

令 $\pi(x)$ 为不大于 x 的素数的个数, 令 $\gamma(x)$ 为不大于 x 的双生素数(即 p 和 $p + 2$ 都是素数)对的个数, 则有 2000 亿以下的素数的分布表

$a(x = a \times 10^8)$	$\pi(x)$	$\pi(x)/x$	$\gamma(x)$	$\gamma(x)/x$	$\pi(x)/\gamma(x)$
1	5 761 455	0.057 6	440 313	0.004 4	13.1
5	26 355 867	0.052 7	1 840 171	0.003 7	14.3
10	50 847 534	0.050 8	3 424 507	0.003 4	14.8
20	98 222 287	0.049 1	6 388 042	0.003 2	15.4
30	144 449 537	0.048 1	9 210 145	0.003 1	15.7
40	189 961 812	0.047 5	11 944 439	0.003 0	15.9
50	234 954 223	0.047 0	14 618 167	0.002 9	16.1
60	279 545 368	0.046 6	17 244 410	0.002 9	16.2
70	324 043 411	0.046 3	19 886 178	0.002 8	16.3
80	368 022 713	0.046 0	22 440 193	0.002 8	16.4
90	411 762 254	0.045 8	24 967 227	0.002 8	16.5
100	455 291 570	0.045 5	27 468 696	0.002 7	16.6
200	882 871 030	0.044 1	51 644 719	0.002 6	17.1
300	1 300 756 481	0.043 4	74 701 741	0.002 5	17.4
400	1 713 131 556	0.042 6	97 206 561	0.002 4	17.6
500	2 121 212 904	0.042 4	119 236 894	0.002 4	17.8
600	2 525 923 387	0.042 1	140 889 354	0.002 3	17.9
700	292 761 714	0.041 8	162 193 472	0.002 3	18.1
800	3 327 476 097	0.041 6	183 349 181	0.002 3	18.1
900	3 725 128 389	0.041 4	204 263 273	0.002 3	18.2
1000	4 121 142 864	0.041 2	225 021 827	0.002 3	18.3
1100	4 515 529 502	0.041 1	245 584 140	0.002 2	18.4
1200	4 908 355 075	0.040 9	265 971 790	0.002 2	18.5
1300	5 299 990 211	0.040 8	286 044 398	0.002 2	18.5
1400	5 690 657 125	0.040 6	306 240 463	0.002 2	18.6
1500	6 080 015 262	0.040 5	326 282 693	0.002 2	18.6
1600	6 468 124 351	0.040 4	346 173 734	0.002 2	18.7
1700	6 855 778 167	0.040 3	366 065 220	0.002 2	18.7
1800	7 242 278 078	0.040 2	385 810 542	0.002 1	18.8
1900	7 628 095 162	0.040 1	405 496 543	0.002 1	18.8
2000	8 013 121 086	0.040 1	425 094 774	0.002 1	18.9

参考文献:

- [1] 华罗庚. 数论导引[M]. 北京: 科学出版社, 1979.
- [2] 柯 召, 孙 琦. 数论讲义[M]. 北京: 高等教育出版社, 1986.
- [3] 曹珍富. 数论中的问题与结果[M]. 哈尔滨: 哈尔滨工业大学出版社, 1996.

Spread of Eratostheness Sieve Method and Distribution of Prime Number

ZHOU Shang-chao, LIU Er-gen

(School of Natural Science, East China Jiaotong University, Nanchang 330013, China)

Abstract: The sieve method of calculation prime number is improved and spreaded for sieve method of stage, which extends range of calculation, raises speed of moving, and calculates prime number and prime number couple under 200 billions.

Key words: prime number; sieve method; Delphi