Studying prime numbers with Maple

Gábor Kallós *

HU ISSN 1418-7108: HEJ Manuscript no.: ANM-000926-A

"To decide, whether a number of 15- or 20-digits is a prime or not, it is not enough even a lifetime, no matter how we use all of our knowledge" – Marin Mersenne, 1644.

Abstract

In this paper we use the Maple [1] software to bring the romantic world of prime numbers closer. We discuss the most important results and their application in the Maple system. Moreover a new, interesting result will be presented: a formula, which produces a lot of primes.

1 Introduction

The investigation of prime numbers has always been a very interesting research field in the history of mathematics. Through many centuries a lot of great mathematicians tried to solve the problems related to these numbers, but in spite of this there have remained even today important open questions.

The theory of prime numbers plays an important role in the university mathematics education, but this field is not easy to demonstrate with the tools of traditional education.

In this paper we present the most important results from prime theory and its applications in Maple. Our purpose here is not to study the whole number theory, the consideration is restricted to this specific field. Thus, sometimes we omit very important related results (such as geometrical connections), or we use some notions without deep preparation (such as congruences, number theoretical funtions). For similar reason, usually the proofs are omitted, most of these can be found in any thorough text-book, e.g. in [4] and [6]. In some cases, where the proof is short and useful, it will be presented. However, there are some results too, the proof of which is extremely difficult. In these cases text-books usually refer to an original source.

Through our discussion we present the short history of this field, with the names of the important authors. More historical results can be found, besides the text-books e.g. in [8], [9] and [11].

The illustration of theoretical results contains Maple examples and short programs. These parts suppose, that the reader has some Maple language expertise. Thorough description of the use of Maple can be found e.g. in [2], [3] and [10].

*Department of Computer Science, Széchenyi István College, H-9026 Hédervári út 3., Győr, Hungary. E-mail:kallos@rs1.szif.hu

[1]Maple is a registered trademark of Waterloo Maple Inc.

Prime numbers and irreducible numbers

We consider in the following integer numbers, i.e. numbers in $\mathbf{Z}$. If a number a is a divisor of b, we denote it by $a|b$.

DEFINITION 1. **Prime property**

For a number $p \neq -1, 0, 1$ let $p|(a \cdot b)$. If from this $p|a$ or $p|b$ follows, then we call the number p a prime number.

DEFINITION 2. **Irreducible property**

We call a number $p \neq -1, 0, 1$ irreducible, if from $p = a \cdot b$ follows $|a| = 1$ or $|b| = 1$.

THEOREM 1.

A number is prime if and only if it is irreducible.

REMARKS

1. This theorem usually does not hold if we leave the ring $\mathbf{Z}$. It is true in every case, that a prime element is irreducible. But the other direction gets violated in many rings. For example in $\mathbf{Z}[\sqrt{8}]$ the number $\sqrt{8}$ is irreducible, but $\sqrt{8}|2 \cdot 4$ does not imply, that $\sqrt{8}|2$ or $\sqrt{8}|4$.

2. Not prime elements are called composite numbers.

THEOREM 2. **The base theorem of number theory**

Every number differ from 0, -1 and 1 can be written as the product of finite irreducible numbers. This factorization is unique regardless of the numbers 1, -1 and the rank of the factors.

In the Maple system we can factorize a number with the function `ifactor`.

```
> ifactor(12345678901234567890123456789O);
```

$$(2)\,(3)^3\,(5)\,(7)\,(13)\,(31)\,(37)\,(211)\,(241)\,(2906161)\,(3803)\,(3607)\,(2161)$$

2 Important properties of prime numbers

Results in the ancient time

THEOREM 3.

The number of primes is infinite.

PROOF. (By EUCLID)

Let us assume, that there are only finite prime numbers, $p_1, p_2, \ldots, p_n$. Let us consider the number $N = p_1 \cdot p_2 \cdot \ldots \cdot p_n + 1$. This number is not divisible by the primes listed above, and the number N is not listed as a prime number, so it cannot be written in the form of Theorem 2. But this is not possible, thus N is a new prime itself, or the product of new primes not listed.

```
> ifactor(2*3*5*7*11*13*17+1);
```

$$(19)\,(97)\,(277)$$

PROPOSITION 1.

If a number n has a prime divisor $p < n$, then it has a prime divisor $p_1 \leq [\sqrt{n}]$.

Using this result, we are able to specify the prime numbers until n, if we know the primes until $[\sqrt{n}]$. The following method is called the sieve of ERATOSTHENES after the author.

Let us write the positive integer numbers beginning with 2 (or 1). Let us consider the first number in the list (we erase and skip the number 1), and erase its multiples. After it we take

the following non-erased number in the list, and repeat the procedure. If we reach a non-erased number greater than $\sqrt{n}$, then the sieving is complete.

After these results there was only very limited development in the prime number theory until the 17th century, FERMAT's time. In the following the most important results of the past four centuries will be summarized.

The number of primes

Let us denote the number of prime elements less-equal to x by $\pi(x)$, where x is a positive real number. From the result of EUCLID follows, that $\pi(x) \to \infty$, if $x \to \infty$. Already EULER noted, that $\lim_{x\to\infty} \frac{\pi(x)}{x} = 0$. This was first proved by LEGENDRE.

The following very important theorem was conjectured in the late 1790s by LEGENDRE and GAUSS, and first proved by HADAMARD and DE LA VALLÉE POUSSIN (1896) with extremely complicated mathematical tools. Elementary proof was given first by PAUL ERDŐS and A. SEELBERG (1948).

THEOREM 4. **The big prime number theorem**

$$\pi(x) \sim \frac{x}{\ln x}.$$

REMARKS

1. The notation $\sim$ means "asymptotic equal" i.e.

$$\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\ln x}} = 1.$$

2. Let us denote the nth prime number by p_n. From the result of the theorem follows $p_n \sim n \ln n$.

3. The approach presented in Theorem 4. can be improved with

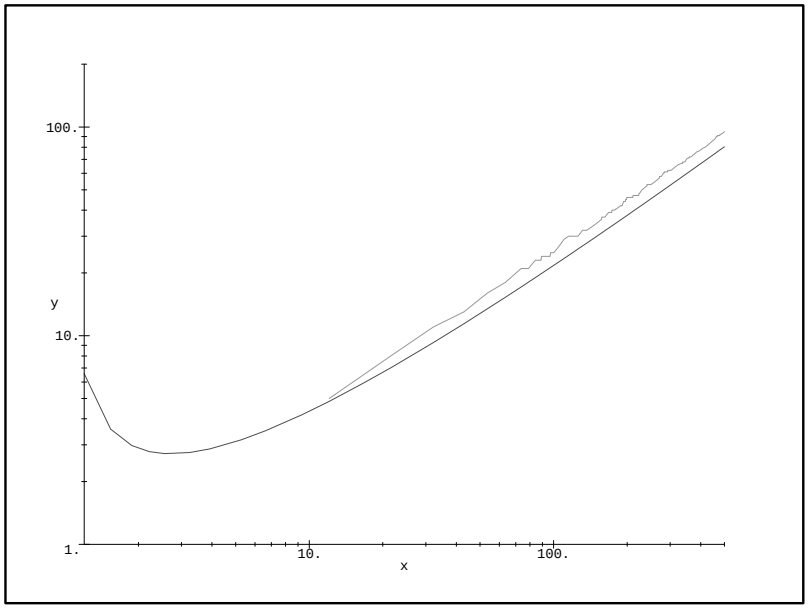
$$\pi(x) \sim \int_2^x \frac{du}{\ln u}.$$

To illustrate the results of the theorem with Maple, we write the function $\pi(x)$, and delineate it with $\frac{x}{\ln x}$ in a logarithmic coordinate-system.

```
> a[2]:=0:
> for i from 3 to 10000 do  a[i]:=a[i-1]:
>   if isprime(i-1) then a[i]:=a[i]+1 fi:
> od:
> pi:=x->if x=trunc(x) then a[x] else a[trunc(x+1)] fi:

> with(plots):
> loglogplot(\{x/ln(x),pi(x)\},x=1.2..500,y=1..200);
```

By studying the figure we can see the stepfunction-like behaviour of $\pi(x)$ (plotted above), and the fact, that for larger numbers both of the functions fit tight the same asymptotic line. However, between two asymptotic equal functions it is allowed to be very large differences.




```
> pi(10000); evalf(10000/ln(10000));
```

$$1229$$

$$1085.736205$$

```
> ""/";
```

$$1.131950831$$

To point 2. of the Theorem, we can produce the ith prime number with the function `ithprime`.

```
> ithprime(1000), evalf(1000*ln(1000));
```

$$7919,\ 6907.755279$$

THEOREM 5. (First proved by EULER)
The sum of the reciprocal value of prime numbers is infinite, i.e. $\sum \frac{1}{p} = \infty$.

REMARKS
1. Comparing with the sum of the reciprocal value of square numbers we get

$$\frac{\pi^2}{6} = \sum \frac{1}{n^2} < \sum \frac{1}{p} = \infty,$$

so the prime numbers are situated more densely.
2. It is true moreover, that

$$\sum_{p \le n} \frac{1}{p} \sim \ln \ln n.$$

Similarly as above, we write the functions $\sum \frac{1}{n^2}$ and $\sum \frac{1}{p}$ to test their behaviour.

```
> sumsquare:=x->sum(1/n^2,n=1..x);
```

$$sumsquare := x \to \sum_{n=1}^{x} \frac{1}{n^2}$$

```
> sumprime:=x->sum(1/ithprime(p),p=1..x);
```

$$sumprime := x \to \sum_{p=1}^{x} \frac{1}{\mathrm{ithprime}(p)}$$

```
> evalf(sumsquare(1000)), evalf(sumprime(1000));
```

$$1.643934568,\ 2.457411277$$

The function $\ln \ln x$ grows very slowly.

```
> evalf(ln(ln(1000))),evalf(ln(ln(1000000)));
```

$$1.932644734,\ 2.625791915$$

The distance of primes

Analysing the result of Theorem 4., we conclude that the primes gradually become rarer and rarer (on the average). Their distribution is nevertheless irregular. For example it is obvious, that the sequence $(n+1)!+2, (n+1)!+3, \ldots, (n+1)!+n+1$ contains no primes, at the same time we know very large consecutive primes.

Let us denote the difference of the nth and $(n-1)$th prime number by d_n, i.e. $d_n = p_n - p_{n-1}$. There are very important results for large and small prime differences.

From Theorem 4. follows that given an arbitrary small number ε there are infinitely many numbers n for which

a) $d_n > (1-\varepsilon)\ln n$,

b) $d_n < (1+\varepsilon)\ln n$.

One of the important results for large prime differences was stated by BERTRAND and proved by CSEBISEV, namely if n is a positive integer, then there is always (at least one) prime number between n and $2n$. We know moreover, that there exists a constant number c, that for every suitable large number n satisfies $d_n < n^c$, e.g. $c = 38/61$ can be written. It is an open question, however, that between two consecutive square numbers falls in every case a prime number.

If $d_n = 2$, than we call the numbers p_n and p_{n-1} twin primes. It is a very old conjecture, that there are infinitely many twin primes. Let us denote the number of twin primes less equal than x by $N(x)$. The most important result here is, that there exists a positive constant c, for which $N(x) < c\frac{x}{\ln^2 x}$. Considering the prime number theorem this means that $N(x)$ is small compared with $\pi(x)$.

To collect observations with Maple about the prime distances we can write programs with the tools reviewed above. Here a simple example will be presented, searching twin primes in a given interval.

```
> twinprimes:=proc(a,limit)
> local i,prev,act;
>   prev:=isprime(a);
>   for i from a by 2 to a+limit do
>     act:=isprime(i);
```

```
>     if prev and act then lprint(i-2,i) fi; prev:=act
>   od
> end:

> twinprimes(10^9+1, 1000);

1000000007    1000000009
1000000409    1000000411
1000000931    1000000933
```

To find the prime neighbours of an arbitrary number we can use the functions `prevprime` and `nextprime`, too.

```
> b:=12345678901234567890:
> prevprime(b), nextprime(b);
```

$$12345678901234567879,\ 12345678901234567891$$

3 Prime formulas

The great mathematicians for centuries were trying to give formulas, which would always produce primes, or at least infinitely many primes. For the second part of this question a nice answer was given by the following theorem, which analyses the occurence of prime numbers in arithmetic sequences.

THEOREM 6. (BY DIRICHLET)
Let a and b be integer numbers, for which gcd$(a, b) = 1$. In this case the sequence $a \cdot k + b$ produces infinitely many primes $(k = 1, 2, \ldots)$.

REMARKS
1. As special cases of Theorem 6., there are infinitely many primes in the form $4k - 1$, $4k + 1$, $6k - 1$, $6k + 1$.
2. We can rephrase the results as follows: the polynomial $ax+b$ with gcd$(a, b) = 1$ produces infinitely many primes. In this context we can formulate some other questions, e.g.
a) Is there a polynomial in the form $ax^2 + bx + c$, which produces infinitely many primes?
b) Is there a polynomial which always produces prime numbers?
In the first case it is easy to prove, that necessary conditions are the irreducibility of the polynomial and gcd$(a, b, c) = 1$, but the complete answer is still unknown. To question b), for

$$p(x) = a_n x^n + \ldots + a_1 x + a_0 = x(a_n x^{n-1} + \ldots + a_1) + a_0$$

if $a_0 = 0$ then $x|p(x)$, if $a_0 \neq 0$ then $a_0|p(a_0)$, so the answer is no. This answer was already given by LEGENDRE.

There are some polynomials, which seem to be good for a lot of substitution values. EULER has presented the example $x^2 + x + 41$, which produces prime numbers for $x = 0, 1, 2, \ldots, 39$, however for $x = 40, 41$ and other larger values we get composite numbers.

```
> p:=x->x^2+x+41:
> for i from 0 to 45 do
>   if not isprime(p(i)) then lprint(i,p(i)) fi:
> od;

40    1681
41    1763
44    2021
```

In spite of the answer given for part b), theoretically is it possible to give a formula which always produces prime numbers. In 1947 H. MILLS proved, that there *exists* a real number A, for which $[A^{3^n}]$ is always prime for an arbitrary positive integer n. However, we do not know the *value* of this number.

Fermat numbers

On the way to find a prime formula, in the 1640s FERMAT drafted, that the numbers $2^{2^n}+1$ are always prime. He established, that beside the small numbers $2^1+1, 2^2+1, 2^4+1$ and 2^8+1 still $2^{16}+1$ is a prime, too. With the number $2^{32}+1$ he and his contemporaries could not reach any result, only in roughly 100 years it was proved by EULER that 641 is a divisor of $2^{2^5}+1$ (with establishing first, that a divisor must be in the form $64k+1$). Nowadays we know moreover, that the Fermat numbers for $n = 6, 7, \ldots 19$ and for more other n values are not primes. We do not know however, if there are another Fermat primes or not.

To produce the Fermat numbers in Maple we use the functions `fermat` or `F` from the package `numtheory`.

```
> with(numtheory):
Warning, new definition for order
> 2^(2^'5')=F(5), ifactor(fermat(5));
```

$$2^{(2^5)} = 4294967297,\ (641)\,(6700417)$$

```
> F(8);
```

$$115792089237316195423570985008687907853269984665640564039457584007\backslash$$
$$913129639937$$

```
> isprime(fermat(8));
```

$$false$$

REMARKS

1. If $k \neq 2^n$, then $N = 2^k+1$ can not be prime. In this case the exponent can be written in the form $k = 2^l \cdot m$, where m is an odd number. Thus

$$N = 2^k+1 = 2^{2^l \cdot m}+1 = (2^{2^l})^m + 1^m = (2^{2^l}+1)((2^{2^l})^{m-1} - + \ldots + 1^{m-1}),$$

i.e. N is divisible by $(2^{2^l}+1)$.

2. The Fermat primes play a very important role in geometry in relation with the construction of regular poligons (GAUSS, see e.g. in [9]).

Mersenne numbers

After a few years of FERMAT's notice, MERSENNE published his examination relating to the numbers in the form $2^s - 1$. He declared, that these numbers are primes if and only if the exponent is $2, 3, 5, 7, 13, 17, 19, 31, 67, 127$ or 257 – assuming that the exponent is less equal than 257 (it was known already in that time, that if $k|s$, then $2^k - 1|2^s - 1$, thus $2^s - 1$ can be only prime, if s is prime). His assertion was not entirely correct, but the first mistake was found only after more than 200 years. É. LUCAS proved in the 1870s that $2^{67} - 1$ is not prime. After this some other mistakes were found in the list.

The numbers in the form above are called Mersenne numbers, and if they are primes, then their names are Mersenne primes. Nowadays we know more than 30 Mersenne primes.

To produce the Mersenne numbers in Maple we use the functions `M` or `mersenne` from the package `numtheory`. The result is the `false` value if the current Mersenne number is not prime. Using the functions with double parentheses ($[i]$) we get exactly the ith Mersenne prime.

```
> mersenne([9]);
```

$$2305843009213693951$$

```
> 2^9-1, M(9);
```

$$511, \mathit{false}$$

```
> M(127);
```

$$170141183460469231731687303715884105727$$

With these tools we are immediately able to find some mistakes in MERSENNE's list.

```
> 2^67-1, mersenne(67);
```

$$147573952589676412927, \mathit{false}$$

```
> M(89);
```

$$618970019642690137449562111$$

REMARK

In the modern history of mathematics the largest known primes were always Mersenne primes. For example from 1772 through a century the number $2^{31} - 1$ (proved by EULER), until 1950 the number $2^{127} - 1$ (proved by É. LUCAS and E. FAUQUEMBERGUE). After this time more Mersenne primes were found with computers, in the past few years since 1985 the largest was $2^{216091} - 1$. From 12.01.1994 the largest is $2^{859433} - 1$.

Perfect numbers

We call a number perfect, if the sum of its divisors – not considering the number itself – is the same as the number (or using the number theoretical function $\sigma(n)$ – the sum of divisors – $\sigma(n)/n = 2$). For example for 6, $1 + 2 + 3 = 6$. Such numbers were already investigated by the ancient Greeks, the notion was first mentioned by PLATON. Until the Middle Ages mysterious properties were atributed to perfect numbers. It is not hard to prove, that an even number is perfect if and only if it is in the form of $2^{t-1} \cdot (2^t - 1)$, where $(2^t - 1)$ is a Mersenne prime. Thus the investigation of perfect numbers is closely related with Mersenne primes. The first 4 perfect numbers were already known by the ancient mathematicians, additional 3 were discovered in the 15-16th century, till MERSENNE's time. We know, that the last digit of an even perfect number must be 6 or 8.

Until now nobody could find any odd perfect number, and it seems, that in the near future we have no hope to prove, that an odd perfect number could not exist. In any case, if such number exists, it must be very large...

To the examination of the perfect numbers with Maple we can use the function `sigma`. Fast "control" can be done with the function `divisors`.

```
> c:=2^12*(2^13-1), sigma(c)-c;
```

$$c := 33550336,\ 33550336$$

```
> divisors(8192);
```

$$\{1,\ 2,\ 4,\ 8,\ 16,\ 32,\ 64,\ 128,\ 512,\ 1024,\ 4096,\ 256,\ 2048,\ 8192\}$$

We call a number k-times perfect, if $\sigma(n)/n = k$ for $k \in \mathbf{Z}$. These numbers were already investigated by MERSENNE, DESCARTES and FERMAT. Nowadays we know already 334 k-times perfect numbers, until the value of $k = 8$ ([4]).

```
> d:=1476304896, sigma(d)/d;
```

$$d := 1476304896,\ 3$$

With the generalization of this notion we can investigate "perfect-like" numbers, where there is a relation between the sum of divisors not considering the number itself ($d = \sigma(n) - n$) and the number n, such as $d/n = a/b$ or $d = n - a$, where a and b are (positive) integers.

```
> e:=3^2*(3^3-1), (sigma(e)-e)/e;
```

$$e := 234,\ \frac{4}{3}$$

We have used a Maple program to produce the following results:

Relation	Numbers
$d = n$	6, 28, 496, 8128
$d/n = 2$	120, 672
$d/n = 3$	30240, 32760
$d/n = 4/3$	12, 234
$d/n = 5/3$	84, 270, 1488, 1638, 24384
$d/n = 7/5$	30, 140, 2480, 6200, 40640
$d = n - 1$	1, 2, 4, 8, 16, ...
$d = n - 2$	3, 10, 136

Some results in this tabular are obvious, e.g. in the line $d = n - 1$ there are pure 2 powers, since $2^k = 1 + 2 + 2^2 + \ldots + 2^{k-1}$.

4 Primality testing and factorization

Given a number n, it is a very important question, how to construct its complete factorization according to Theorem 2. Up to the 1950s, this process was usually very circumstantial – and in most of the cases unsuccessful – for large numbers. In spite of this a number of very important theoretical results were published already long before (e.g. by FERMAT). However, the significant development in this field practically started with the appearance of computers. Similarly

as before, we present only the most important results and only some of the proofs. We omit the exact behaviour-analysis by the methods. Detailed theoretical discussion can be found e.g. in [1].

The basic algorithm

A traditional method for the factorization is the trial divison with the primes $2, 3, 5, 7, 11, 13, 17, \ldots$, or with the numbers $2, 3, 5, 7, 9, 11, 13, 15, \ldots$, which is easier to programize, but we get the result slower. Analysing this algorithm (e.g. its simplified Maple-form below) we conclude, that it works usable in our PC's if the order of n is maximum $10^{16} - 10^{18}$, or if the order of the divisor is maximum $10^8 - 10^9$. For larger numbers the algorithm usually slows down hopelessly, because of the lot of divisions. In a faster computer we can make this situation better, but the improvement will not be spectacular.

In the following example we search for the prime divisors of an odd number with trial divison. We can get the same result using the `ifactor` function with the option `'easy'`.

```
> basic:=proc(n,limit)
> local i,s,a;
>   s:=NULL; a:=n;
>   for i from 3 by 2 to limit do
>     if a mod i = 0 then a:=a/i; s:=s,i;
>       while a mod i = 0 do a:=a/i; s:=s,i od
>     fi
>   od; print(s,a)
> end:

> b:=12345678901234567890:
> basic(b, 4001);
```

$$3,\ 3,\ 101,\ 3541,\ 3607,\ 3803,\ 27961$$

```
> ifactor(b,'easy');
```

$$(3)^2\,(101)\,(3803)\,(3607)\,(27961)\,(3541)$$

Let us assume, that the number n is very large, it contains e.g. 100 decimal digits. In this case we apply the basic algorithm to separate the "small" factors up to the order of approximately 10^9. After this we examine the remaining part further with additional methods, discussed in the following subsections.

4.1 Prime tests

Let n be an odd number after the separation of the small factors. First we have to decide, if this number is prime or not.

A test based on Fermat's little theorem

Already FERMAT has proved the following very important theorem:

THEOREM 7. FERMAT**'s little theorem**
If p is a prime which does not divide a, then $a^{p-1} \equiv 1 (\mathit{mod}\ p)$.

REMARK

This theorem is a special case of a theorem of EULER, which asserts for relatively prime integers a and m, that $a^{\phi(m)} \equiv 1 (\text{mod } p)$, where the function $\phi(n)$ gives the number of the relative prime positive integers less equal to n.

FERMAT's theorem gives a very effective tool to filter out most of the composite numbers. If e.g. $2^{n-1} \not\equiv 1 (\text{mod } n)$, then we know surely, that the number n is composite.

```
> 2^118 mod 119;
```

$$30$$

Though the computing of these powers seems to be difficult, there are very efficient and fast algorithms for this ([1], [8]).

In Maple we can use the more efficient `modp` and `power` functions instead of the traditional power operator. Thus, we can prove immediately, that the 5th Fermat number is not prime.

```
> 3^(2^32) mod (2^32+1);
Error, integer too large in context
> modp(power(3,2^32),2^32+1);
```

$$3029026160$$

Unfortunately, the reverse of the theorem does not hold entirely. So if we get $2^{n-1} \equiv 1 (\text{mod } n)$, then in spite of this is it possible for n to be a composite number. Such number n is called a pseudoprime (in base 2). The pseudoprimes are fairly rare, analysing the chance of their occurence compared to the primes we get at most a few per thousand.

```
> pseudo:=proc(n)
> local i,s;
>   s:=NULL;
>   for i from 3 by 2 to n do
>     if not(isprime(i)) and (2^(i-1) mod i)=1 then s:=s,i
>     fi
>   od;
>   RETURN(s)
> end:

> s1:=pseudo(3001)
```

$$s1 := 341, 561, 645, 1105, 1387, 1729, 1905, 2047, 2465, 2701, 2821$$

Moreover, additional pseudoprimes can be unveiled choosing another base instead of 2, such as $3, 5, 7, \ldots$.

```
> 3^1386 mod 1387, 1387=ifactor(1387);
```

$$875, 1387 = (19)\,(73)$$

There are however such extreme numbers (called Carmichael numbers), which are pseudoprimes in all bases, which are relative prime to all of their divisors. These numbers are very rare, the chance of choosing such a (not very small) number is less than approximately one to a

million. The first such number is 561. The Carmichael numbers have at least 3 different prime divisors, so one of their divisors is less than $\sqrt[3]{n}$ ([8]).

We can search the Carmichael numbers with the following small program, the input parameter of which is a sequence of pseudoprimes.

```
> carmic:=proc(s)
> local s2,i,n,j,carm;
>   s2:=NULL;
>   for i from 1 to nops([s]) do
>     carm:=true: n:=trunc(sqrt(s[i])):
>     for j from 3 by 2 to n do
>       if modp(power(j,n-1),n)<>1 and (n mod j)>0 then
>       carm:=false fi
>     od;
>     if carm then s2:=s2,s[i] fi
>   od;
>   RETURN(s2)
> end;

> carmic([s1]);
```

$$561,\ 1387,\ 1729,\ 1905,\ 2821$$

If a number n passes through the filters $2, 3, 5$ and 7 then it is recommended to stop this method. There are some other methods to unveil such composite numbers.

A strong pseudoprime test

Let us assume, that $a^{n-1} \bmod n = 1$, where $n = 2i - 1$ (n is odd), and $\gcd(a, n) = 1$. In this case n is a divisor of $a^{2i} - 1 = (a^i - 1) \cdot (a^i + 1)$. If n is a prime number, then it divides exactly one of the factors (else it would divide the difference of the factors too), thus $a^i \bmod n = 1$ or $a^i \bmod n = -1$. However, if n is not prime, then we have a good chance, that some divisor of n divides $a^i - 1$, and an another divides $a^i + 1$. Thus we get for the remainder $a^i \bmod n \neq |1|$, since n does not divide any of the factors. We can continue in the same manner with $i = 2j$. Eventually, we are able to filter out most of the pseudoprimes.

```
> n:=1387:
> 2^'1386' - 1 = (2^'693'-1)*(2^'693'+1);
```

$$2^{1386} - 1 = (2^{693} - 1)\,(2^{693} + 1)$$

```
> modp(power(2,693),n);
```

$$512$$

The Miller-Rabin test

The following random method is very likely to be the best tool for primality testing. It was composed and analysed by G. L. MILLER and M. O. RABIN in the late 70s.

Let $n = 1 + 2^k j$ be a prime number, and let $x^j \bmod n \neq 1$, where x is an integer with $1 < x < n$. In this case $x^{2^k j} \bmod n = 1$. Since n is prime, thus $n | (x^{2^{k-1} j} - 1)$ or $n | (x^{2^{k-1} j} + 1)$,

i.e. for $x^{2^{k-1}j}$ mod n we get 1 or -1. Stepping back similarly we get eventually, that in the remainder sequence $r_1 = x^j \bmod n$, $r_2 = x^{2j} \bmod n$, $r_3 = x^{4j} \bmod n, \ldots, r_i = x^{2^k j} \bmod n$ the last value must be 1, and before it we get $n-1$. If the number n is composite, then this probably does not hold.

Thus the algorithm proceeds as follows: we choose a random x with $1 < x < n$. We produce the remainder sequence as above. If this sequence ends with $r_{i-1} = n-1$ and $r_i = 1$, then the number is (likely) prime. If $r_i = 1$ and $r_{i-1} \neq n-1$ then the number is (surely) not prime.

We illustrate the operation of this method to unveil a Carmichael number.

```
> n:=1729:
> ifactor(n-1);
```

$$(2)^6\,(3)^3$$

```
> j:=3^3:
> seq(2^(2^i*j) mod n,i=0..6);
```

$$645,\ 1065,\ 1,\ 1,\ 1,\ 1,\ 1$$

If the algorithm says, that a number is not prime, then this is surely true, and if the answer is prime, then this is not yet certain. M. O. RABIN has proved, that for arbitrary n the algorithm gives wrong answer with a probability at most $1/4$ ([8]). So choosing new random x bases, after e.g. 20 executions, the probability, that the answer is still prime and the number is in spite of this composite, is less than $(1/4)^{20}$. Thus, in practice we can say, that this answer is correct (the exact proof of the prime property is discussed in a subsequent subsection). The significance of this method is, that we get a reliable answer in relatively short time for very large numbers (containing several hundred digits), too.

The `isprime` function in Maple uses this method too ([3]), for such numbers, which seem to be primes with methods described earlier.

4.2 Factorization methods

In this subsection we discuss, how we can find the prime factors of a large number n (knowing that n is not prime). If n contains e.g. more than 30 digits the use of the basic algorithm is usually hopeless, we do not like a lot of time to wait. In the last decades more algorithms were established to improve this situation. However, in comparison with the prime tests, the factorization algorithms are much more expensive. It is much more difficult to factorize a number, than to decide the prime property.

The Pollard-ρ method

One of the well-known methods was presented by J. M. POLLARD. It was called by him a Monte Carlo method, because of its (pseudo) random nature. In spite of this the algorithm terminates usually succesful, and it is easy to programize.

Let x_0 be an integer, and let $f(x)$ be a polynomial with integer coefficients. Let us consider the sequence $x_{i+1} = f(x_i) \bmod n$. Our purpose is, that this sequence must necessarily be random-like. This property depends on the choice of the polynomial $f(x)$. It is proved, that a linear polynomial $ax+b$ is not good, the next simplest case is x^2+1. This choice behaves nice, though we can not prove this exactly.

```
> x[0]:=2: n:=1387: s:=NULL:
> for i from 1 to 8 do x[i]:=x[i-1]^2+1; s:=s,x[i] mod n
> od:
> print(s);
```

$$5,\ 26,\ 677,\ 620,\ 202,\ 582,\ 297,\ 829$$

Let us assume, that the number n has a non-trivial divisor d. Considering the sequence $y_i = x_i \bmod d$ we find, that this sequence will be eventually periodic (there are only finite different residues mod d). The path of the y_i-s draws a greek letter ρ, a tail with a cycle. That is why this algorithm is called as the Pollard-ρ method. We get $y_j = y_k, y_{j+1} = y_{k+1}, \ldots$, for some indices $j \neq k$. Using this $x_j \equiv x_k \pmod d$, i.e. $d|x_j - x_k$, thus $\gcd(n, x_j - x_k)$ is a non-trivial divisor of n. We do not know the value d, however if we choose a lot of pairs (j, k), and compute for the pairs $\gcd(n, x_j - x_k)$, then usually sooner or later we will find a factor of n. The efficiency of this algorithm can be improved if we compute the gcd-s rarer, only for products e.g. 10 pairs of $x_j - x_k$ and n. To compute integer gcd-s, usually some version of the Euclidean algorithm (see e.g. in [5]) is used, which is relatively fast.

We illustrate the theoretical description with the factorization of the number 1387. After finding a divisor we produce the y_j values and present the "ρ property".

```
> l:=[s]: prod:=1:
> for i from 2 to 8 do prod:=prod*(l[i]-l[i-1]) od:
> prod, igcd(prod,n);
```

$$-18766855483437600,\ 19$$

```
> for i from 1 to 8 do l[i]:=l[i] mod 19 od: l;
```

$$[5,\ 7,\ 12,\ 12,\ 12,\ 12,\ 12,\ 12]$$

```
> ifactor(1387);
```

$$(19)\,(73)$$

Getting a divisor d, we check the numbers d and n/d. If we find that these numbers are not primes, we can try to factorize them further with the same method or with the basic algorithm.

It is not sure, that the algorithm gives a correct answer. It is possible, that the gcd is n. In this case the choice for $f(x)$ seems to be wrong, let us choose another polynomial $f(x) = x^2 + c$, $c \neq 0, 1, -2$. If there is no answer after a long time, then it is better to try another factorization algorithm.

With Maple we can apply this method to factorize integers, using the function `ifactor` with the option `'pollard'`.

Pollard's $p-1$ method

Another method of POLLARD is based on the assumption, that the number n has such a prime factor p, for which $p-1$ is a product of small primes. If this happens, then for an arbitrary number a with $\gcd(a, p) = 1$, the equation $a^{p-1} \equiv 1 \pmod p$ is satisfied. Thus $p|\gcd(a^{p-1} - 1, n)$. However, to find this divisor p we have to be lucky. Let us choose a number $i = 2^{c_2} \cdot 3^{c_3} \cdot 5^{c_5} \cdot \ldots \cdot k^{c_k}$, where the bases are the first few primes, and the exponents are small positive integers. After it we compute $\gcd(a^i - 1, n) = d$. If $n > d > 1$, then we have found non-trivial factors of n, d and n/d. If $d = 1$, then we have to choose a larger i, and if $d = n$, then we need another base a. With this method (called Pollard $p-1$) we find sometimes very quickly a non-trivial divisor, but in many cases it does not work.

```
> n:=973: i:=2^5*3^3*5*7:
> igcd(2^i-1,n);
```

$$7$$

```
> ifactor(n);
```

$$(7)\,(139)$$

By combining the basic algorithm with the Pollard methods in most of the cases we are able to factorize numbers up to roughly $30-40$ (sometimes 50) decimal digits. There are some other algorithms, which are suitable for factorization of such numbers, e.g. a method developed by FERMAT (see [1] or [8]). The adventage of his idea is, that this algorithm uses only additions and substractions, there are no divisions. However, to a significant improvement we need other approaches.

Sieving algorithms

With the improvement of FERMAT's original concept we get sieving algorithms ([8]). In the beginning these methods yield not a significant improvement over 50 decimal digits. However, in the early 80s the quadratic sieve was discovered, and after a few years with further developments (first of all by POMERANCE) it proved to be one of the best tools to decompose large numbers ([1]). In a decade it almost entirely displaced the formerly successful continued fraction algorithm.

An elliptic curve method

Besides the quadratic sieve nowadays the most successful factorization algorithm is the really surprising elliptic curve method worked out by H. W. LENSTRA. In this paper we only outline this method, because the detailed discussion needs longer introduction to the elliptic curve theory. Its basic idea is similar to the Pollard $p-1$ method. So very briefly, we choose an elliptic curve in the normal form $y^2 = x^3 + bx + c$, and a number p similarly as in the algorithm Pollard $p-1$. We work with a finite group of points (generated by a number p) on this curve, and compute a gcd. However, comparing with the Pollard method, if this gcd is 1, then we have the possibility to choose another ellipitic curve. Because of the significant difference of the groups (generated by the same number p) on different elliptic curves, sooner or later we have a very good chance to find a non-trivial divisor (details in [1] or [12]).

With Maple we can apply this method using the function `ifactor` with the option `'lenstra'`.

REMARK

The factorization of large numbers can seem a useless or only theoretically interesting thing. However, a very important practical application was discovered in the 70s: a public key cryptosystem. The essence of the method is, that the encrypted message and the key are public, but in spite of this unauthorized persons are not able to read the message, because to do this it is needed to factorize a very large number, which is the product of two large primes (detailed discussion in [1]). If our "enemy" tries to discover the message, using a 250 digit key, it takes years, even if he or she uses the best current supercomputers and algorithms.

4.3 Exact proof of the prime property

With the methods discussed above we can prove very fast – in most of the cases –, that a number is composite, but usually we are not able to prove exactly the prime property (even using our best

tool, the Miller-Rabin test). To this we have before this section only the basic algorithm, which is useless for larger numbers. Fortunately, we have some other methods which are relatively easy to programize and are efficient.

We call a number g primitive root (mod p), if $g^{p-1} \bmod p = 1$ and $g^k \bmod p \neq 1$ for $1 \leq k < p-1$. If there exists a primitive root (mod p), then p is a prime number, since in this case the numbers $g^k \bmod p$ for $1 \leq k \leq p-1$ are all different and produce the numbers $1, 2, \ldots, n-1$ in some order of succession, i.e. p does not have a non-trivial divisor. The number of the primitive roots (mod p) is $\phi(p-1)$, so it is relatively easy to find such a number. For example 2 is a primitive root modulo 11.

```
> seq(2^i mod 11,i=1..10);
```

$$2,\ 4,\ 8,\ 5,\ 10,\ 9,\ 7,\ 3,\ 6,\ 1$$

After finding a number x for which $x^{n-1} \bmod n = 1$, we know already, that the order of x is a divisor of $n-1$. If the order is exactly $n-1$, then n is prime. So we have to check the exponents, which are the (prime) divisors of $n-1$, namely the exponents $(n-1)/p$.

Thus, by the check of the following two conditions we have a very nice method to prove exactly the prime property:

a) $x^{n-1} \bmod n = 1$,

b) $x^{(n-1)/p} \bmod n \neq 1$, for all prime divisors p of $n-1$.

It is known moreover, that the x bases here are allowed to be different, the prime property is still satisfied (J. BRILLHART, D. H. LEHMER and J. L. SELFRIDGE, 1975), details in [1]. Thus to prove the prime property of n, we need the complete factorization of $n-1$. However in many cases we are not able to factorize $n-1$, if n is very large.

That is, why perfected methods were worked out, using only the partial factorization of $n-1$ or that of $n+1$.

In the first case let us write n in the form $n = d_1 \cdot d_2 + 1$, where $0 < d_2 \leq d_1 + 1$. If for all prime divisor p of d_1 there exists an x, for which $x^{n-1} \bmod n = \gcd(x^{(n-1)/p} - 1, n) = 1$, then n is prime (H. C. POCKLINGTON, 1914).

We use these two methods in the following subsection to prove the prime property of a large number.

Similarly as by the factorization, there are exact prime tests using elliptic curves (details in [1]). Moreover, very effective methods are known to prove the prime properties of numbers in special forms, e.g. for Fermat numbers and Mersenne numbers. These methods start so, that we know the factorization of $n-1$ or $n+1$, but they use further special improvements. The following test for Mersenne numbers was worked out by É. LUCAS and D. H. LEHMER (proof in [8]).

Let p be an odd prime, and let us define the sequence $\{L_i\}$ in the following manner: $L_0 = 4$, $L_{i+1} = (L_i^2 - 2) \bmod (2^p - 1)$. Then $2^p - 1$ is prime if and only if $L_{p-2} = 0$.

Due to this method, we are able to prove the prime property of very large Mersenne numbers. Here we check the 13th Mersenne number.

```
> L:=4: l:=4: m:=13:
> for i from 1 to m-2 do
>   L:=(L*L - 2) mod (2^m-1): l:=l,L
> od: l;
```

$$4,\ 14,\ 194,\ 4870,\ 3953,\ 5970,\ 1857,\ 36,\ 1294,\ 3470,\ 128,\ 0$$

4.4 Application in generalized Pascal's triangles

In this section we apply the methods discussed in the previous subsection to prove exactly the prime property of a large number in a generalized Pascal's triangle. The theory of these triangles was presented by the author in 1997 ([7]). The general triangles were specified as follows:

DEFINITION. (generalized Pascal's triangle)

Let $0 \leq a_0, a_1, a_2 \ldots a_{m-2}, a_{m-1} \leq 9$ be integers. Then we can get the kth element in the nth row of the $a_0a_1a_2 \ldots a_{m-2}a_{m-1}$-based triangle if we multiply the $(k-m)$th element in the $(n-1)$th row by a_{m-1}, the $(k-m+1)$th element in the $(n-1)$th row by a_{m-2}, ..., the $(k-1)$th element in the $(n-1)$th row by a_1, the kth element in the $(n-1)$th row by a_0, and add the products. If for some i we have $k-m+i < 0$ or $k-m+i > n-1$ (i.e. some element in the $(n-1)$th row does not exist according to the traditional implementation) then we consider this element to be 0. The indices in the rows and columns of the triangle run from 0.

In [7] the most important properties of these triangles were thoroughly investigated. Among others we gave a direct formula for the kth element in the nth row, thus we are able to specify an element somewhere in a generalized triangle without building the preceding rows.

Investigating the elements in these triangles we can find a number of large primes. It is a conjecture, that we are able to find *arbitrary* large primes too, so with the building of the triangles we get some kind of "prime formula".

For example in the first 150 rows in the 1114 based triangle with a Maple program using the `isprime` function we found the elements in the following (row, column) position to be prime:

$$(2,2), (3,3), (3,6), (4,4), (4,8), (5,5), (10,10), (23,46), (31,62),$$

$$(48,48), (116,116), (132,132), (144,144).$$

Here the last element contains 93 decimal digits. In Figure 1 we present the first few row of this triangle.

$$\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & \mathbf{1} & \mathbf{1} & 4 & & & & \\
 & & & & 1 & 2 & \mathbf{3} & 10 & \mathbf{9} & 8 & 16 & & \\
 & & & 1 & 3 & 6 & \mathbf{19} & 30 & 39 & \mathbf{73} & 60 & 48 & 64 \\
1 & 4 & 10 & 32 & \mathbf{67} & 112 & 218 & 292 & \mathbf{337} & 464 & 352 & 256 & 256 \\
 & & & & & & \ldots & & & & & &
\end{array}$$

Figure 1: The 1114-based triangle

Analysing the structure of the $abcd$ based triangle, we conclude, that prime elements can *only* take place in the positions (n,n) and $(n,2n)$ – this follows from Theorem 2 in [7]. The elements in these positions are presented with bold characters.

We mentioned above, if the `isprime` function says, that a number is prime, it is not surely true. Thus if we would like to make sure about this property, we have to use the methods discussed in the previous subsection.

Exact proof for a prime candidate

In the following part we prove exactly the prime property of the element in the position $(116,116)$, which is

```
> a:=113339208484022902352005233461239034435177252913302015
> 128561961100032061289;
```

a number containing 75 digits. Testing this number with the `isprime` function we get

```
> isprime(a);
```

$$true$$

thus a is very likely prime. To prove this, we need the factorization of $a-1$. We use first the `easy` option, because else we can run in a very long unsuccessful cycle.

```
> ifactor(a-1,'easy');
```

$$(2)^3\,(3)\,\_c69\,(5399)$$

```
> b:=(a-1)/(2^3*3*5399);
```

$$b := 87469290982915742384396210302246584579842912972542766506576\8\backslash$$
$$051954313$$

```
> isprime(b);
```

$$false$$

Knowing that b is not prime, we have a 69-digit number to factorize. It is usually hopeless in a PC, but now we are lucky, with the `pollard` option we can break this number in approximately 15 minutes [2]. Without this option, with the basically interpreted Morrison-Brillhart method, the `ifactor` will stay unsuccesful for much longer time.

```
> ifactor(b,'pollard');
```

$$(10819096345425007095734545669640557980594988462603498411463)$$
$$(80847131951)$$

```
> c:=b/80847131951:
```

Observing, that the Maple-system writes the complete factorization only if the factors are prime (in the other case e.g. for c we would have *_c59*), we know, that investigating c with the `ifactor` function we get the answer *true*. However, we have to prove this too. The question arises, when can we believe "surely" the Maple-system if it says, that a number is prime. Analysing the operation of the `isprime` function with the following commands [3]

```
> interface(verboseproc=2):
> print(isprime);
```

[2] Here and below: using at least a PC Pentium

[3] The output of the print function is omitted

we conclude, that the system examines gcd-s up to approximately 10^6, so very strictly this is the lower bound. However, the Maple guarantees the safety for much larger numbers, too. In this paper we fix our lower bound – considering the point of view of safety and decreasing the use up of space – at 20 decimal digits. Thus we believe the *true* answer of the `isprime` function without reservation up to this bound (it is an easy exercise to check, that the *prime* numbers below this bound are really primes). Turning back to the number c, we have to factorize $c-1$.

```
> ifactor(c-1,'easy');
```

$$(2)\,(23)\,(617)\,(3811956995780779048599304372362961729474662977451729 41)$$

```
> d:=(c-1)/(2*23*617):
```

Now we have a 54 digit large prime candidate, let us continue with $d-1$.

```
> ifactor(d-1,'easy');
```

$$(2)^2\,(5)\,(7)^2\,(11)\,(13)\,(29)\,(853)\,\_c35\,(551231)\,(17977)$$

```
> e:=(d-1)/(4*5*49*11*13*29*853*17977*551231);
```

$$e := 11096550617640991328150412126833359$$

We know surely, that this 35 digit number is not prime. The Maple is able to factorize it in 15 minutes.

```
> ifactor(e);
```

$$(51386119357470140587)\,(215944514907757)$$

So finally we have a complete factorization, thus we begin the exact proofs. First we prove the prime property of d with the method of POCKLINGTON.

```
> f:=215944514907757:
> g:=51386119357470140587:
> modp(power(2,d-1),d);
```

$$1$$

Observing, that $f \cdot g > [\sqrt{d}]$ we have to check only two gcd-s.

```
> gcd(modp(power(2,(d-1)/f),d),d);
```

$$1$$

```
> gcd(modp(power(2,(d-1)/g),d),d);
```

$$1$$

Thus d is prime, we can continue with c. Now we use the method of BRILLHART, LEHMER and SELFRIDGE, because $c-1$ has only a few prime divisors.

```
> modp(power(2,(c-1)/d),c);
```

$$5070659633100738759368526184715994149200912905277760631436$$

Similarly we get very large results for `modp(power(2,(c-1)/617),c)` and `modp(power(2,(c-1)/23),c)`. Here we disregarded the details. With our last exponent first we are unsuccessful.

```
> modp(power(2,(c-1)/2),c);
```

$$1$$

Choosing another base we get

```
> modp(power(3,(c-1)/d),c);
```

$$35777904115108435185209606901253904274812325596548050863 66$$

thus we are ready with c. To the prime property of a, we observe, that c is much larger, than the product of the other factors of $a-1$, i.e. with the POCKLINGTON method we need to compute only one gcd.

```
> modp(power(2,a-1),a);
```

$$1$$

```
> gcd(modp(power(2,(a-1)/c),a),a);
```

$$1$$

With this we have proved exactly the prime property of a.

Conclusions

Summarizing, we have found very large primes in the generalized Pascal's triangles, and the Maple system was able to prove exactly the prime property of one candidate.

References

[1] DAVID M. BRESSOUD, *Factorization and Primality Testing*, Springer Verlag, New York, 1989.

[2] B. W. CHAR, K. O. GEDDES, G. H. GONNET, B. L. LEONG, M. B. MONAGAN, S. M. WATT, *Maple V Language Reference Manual*, Springer Verlag, New York, 1991.

[3] B. W. CHAR, K. O. GEDDES, G. H. GONNET, B. L. LEONG, M. B. MONAGAN, S. M. WATT, *Maple V Library Reference Manual*, Springer Verlag, New York, 1991.

[4] ERDŐS PÁL, SURÁNYI JÁNOS, *Válogatott fejezetek a számelméletből*, Polygon, Szeged, 1996.

[5] K. O. GEDDES, S. R. CZAPOR, G. LABAHN, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1991.

[6] GYARMATI EDIT, TURÁN PÁL, *Számelmélet*, Tankönyvkiadó, Budapest, 1988.

[7] GÁBOR KALLÓS, The Generalization of Pascal's Triangle from Algebraic Point of View, *Acta Acad. Paed. Agriensis*, Nova Series Tom. XXIV. (1997). 11-18.

[8] D. E. KNUTH, *The Art of Computer Programming, Vol. 2.*, Addison-Wesley, 1981.

[9] EDWARD KOFLER, *Fejezetek a matematika történetéből*, Gondolat, Budapest, 1965.

[10] MOLNÁRKA GY., GERGÓ L., WETTL F., HORVÁTH A., KALLÓS G., *A Maple V és alkalmazásai*, Springer, Budapest, 1996.

[11] OYSTEIN ORE, *Number Theory and its History*, McGraw–Hill Book Company Inc., New York, 1948.

[12] JOSEPH H. SILVERMAN, JOHN TATE, *Rational Points on Elliptic Curves*, Springer Verlag, New York, 1992.