U.P.B. Sci. Bull., Series A, Vol. 68, No. 3, 2006

COMPUTERISED ALGEBRA USED TO CALCULATE x^n COST AND SOME COSTS FROM CONVERSIONS OF P-BASE SYSTEM WITH REFERENCES OF P-ADIC NUMBERS FROM Z_p AND Q_p

C.A. MURESAN[*]

Autorul acestui articol încearca sa sublinieze importanta utilizarii unor cât mai eficienti algoritmi de calcul pentru ca sa fie cât mai putin utilizate resursele calculatorului, astfel încât numarul de operatii sa fie mai mic.Dupa prezentarea unor algoritmi diferiti, se calculeaza matematic costul algoritmilor inclusiv laconversia din baza p în baza 10 a unor numere p-adice din ***Zp*** *respectiv* ***Qp***.

The author of this article tries to outline the importance of using as much as possible the calculation algorithms in order to reduce the use of the computer. After presenting some different algorithms, the cost of the algorithm is mathematically calculated, including the conversion from a p-adic number into a base-10 system of a number from ***Zp*** *and* ***Qp*** *also.*

Keywords: costs, algorithms, p-adic number

Introduction

The article will focus on two algorithms used for finding out x^n .

In one situation x^n can be directly calculated from the following relation

$x^n = x \cdot x \cdot x \cdot x ....\cdot x$ of 'n' times, whereas in the second one, we will use the binary writing of '*x*'.

Defining the cost of the algorithm as being the number of multiplications used during the calculation of x^n, we will observe that the second algorithm has a cost

$C(2) = a \cdot [\lg_2 n]; a \in [1,2]$; $C(2)$ is the cost ; whereas the first one has a n-1 cost. The second algorithm will be 'faster' than the first one.

In order to demonstrate these statements we must observe that if $n = \sum_{i=0}^{k} e(i) \cdot 2^i; \; e(i) \in \{0,1\}$ is a binary writing of '*n*' then

[*] PhD Student, Dept. of Mathematics, University "Politehnica" of Bucharest, ROMANIA

$$C(2) = k-1+e(0) + e(1) + \ldots + e(k)$$

where $C(2)$ is the cost of the second algorithm.

Passing then to some inequalities we will obtain the preceding statements.

Finally, using the previous formula, we have calculated the conversions' costs of natural numbers in base-p or of some p-adic rational numbers from $\boldsymbol{Q_p}$.

1. Estimates of the costs of the algorithms used to calculate x^n

Definition 1: The cost of an algorithm that calculates x^n is given by the number of the multiplications effectuated until we get the result; if we use number '(i)' to represent the algorithm, then its cost will be *C(i)*.

Method no 1 : x^n calculation using algorithms (1):

```
Data: x,n {y is the result}
y: = x
for i= 1 to n-1 results y = yx
```

Proposition 1: The cost of the algorithm (1) is n-1 meaning that $C(1)= n-1$.

Proof: It is obvious that for finding out x^n using multiplications of x we have $x \cdot x \ldots \cdot x$ for 'n' times which mean that we will have n-1 multiplications.

Method no 2: We can find out x^n using the algorithm (2) where 'n' have a representation in the 2-base system.

Be it $n = \sum_{i=0}^{k} e(i) \cdot 2^i;\ e(i) \in \{0,1\}$ then $x^n = x^{\sum_{i=0}^{k} e(i) \cdot 2^i} = \prod_{i, e_i \neq 0} x^{2^i}$ and here we have the algorithm (2) :

```
Data : x,n
z:= x;y = 1{the result is y}
y: = x
while n >1 do
  begin
  p:= [n/2];
  if n>2*p then y: = y*z;
  z: = z*z
  n:=p
end;
y:=y*z;
```

In this algorithm in 'z' we will obtain the values x^2, x^4, x^8...... for every step, and $y=y*z$ for every step when *'n'* is an even number and *'y'* is the same number when *'n'* is an odd number. Finally *'y'* is x^n.

Example : Find x^{21} (see the next table) using the algorithm

n	21	21	10	5	2	1
p		10	5	2	1	
z	x	x^2	x^4	x^8	x^{16}	
y	1	x	x	x^5	x^5	x^{21}

The multiplications in this algorithm are:

$x^2 = x\cdot x$, $x^4 = x^2 \cdot x^2$, $x^8 = x^4 \cdot x^4$, $x^5 = x^4 \cdot x$, $x^{16} = x^8 \cdot x^8$, $x^{21} = x^{16} \cdot x^5$

Numbers of multiplications for this example is 6 so $C(2) = 6$.

Observation:

If $\overline{(e(n)e(n-1)......e(1)e(0))}_{(2)}$ is the number '21' in 2-base system then because:

21= $1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$ we have

$e(0)=1$,$e(1)=0$,$e(2)=1$,$e(3)=0$, $e(4)=1$ and

$\overline{(e(n)e(n-1)......e(1)e(0))}_{(2)} = \overline{10101}_{(2)}$

$k=4$ where $k+1=5$ is the length of '21' in 2-base system.

Then k-1+e(0) + e (1) + ...+ e (k) =4-1+1+0+1+0+1=6

This is the formula for the algorithm's cost.

Definition 2. .Let $(a_n)_{n\ge 1}$,$(b_n)_{n\ge 1}$ be two sequences of positive numbers.We say that (a_n) *is* $O(b_n)$ if there exist two positive constants c,C such that $c\cdot b_n \le a_n \le C\cdot b_n$ for every $n \in$ N. So, $O\,(b_n) = a_n$.

(O is called' Landau's symbol).

Proposition 2 . $C(2) = O([\lg_2 n])$ and $[\lg_2 n] \le C(2) \le 2[\lg_2 n]$

Proof: $C(2)$ is the number of multiplications used in the algorithm.

From $x^n = x^{\sum_{i=0}^{k} e(i)\cdot 2^i}$, $e(i) \in \{0,1\}$.

we obtain :

$x^n = x^{e(0)\cdot 2^0} \cdot x^{e(1)\cdot 2^1} \cdot .... x^{e(i)\cdot 2^i} \cdot .... \cdot x^{e(k)\cdot 2^k}$

where ’k-1’
will be the number of multiplications for :

$$x^2, x^4 \ldots\ldots x^{2^k}, where 2^i = \left(2^j\right)^2 \ and \ x^{2^i} = (x^{2^j})^2 \ ., \quad for\ i = 2j \quad i, j \in \mathbf{N}.$$

or in fact ,the numbers of multiplications for:

$$x^2 = x \cdot x\ ,\ x^4 = x^2 \cdot x^2\ ,\ x^8 = x^4 \cdot x^4\ ,\quad ,\ x^{2l} = x^l \cdot x^l \ldots\ldots x^{2^k} = (x^{2^{k-1}})^2$$

from the algorithm.

And $\sum_{i=0}^{k} e(i)$;

will be the total number of multiplications between all the terms with the form: $x^{e(i)*2^i}$ $and\ x^{e(j)*2^k}$, $0 \le i, j \le k$

(in order to determine *x).*

Then C(2) = k-1+e(0) + e (1) + ...+ e (k)

We can suppose $e(k) = 1$ without restricting the generality .

But 0+0+0+…..+0+1=e(0) + e(1) +...+ e(k) = 1+1+1...+1= k+1

and then for the two relations results: $k = C(2) \le (k-1) + (k+1) = 2k$.,

where $n = \sum_{i=0}^{k} e(i) \cdot 2^i$; $e(i) \in \{0,1\}$ is the notation for ,n’ in 2-base system

(meaning that ,n’ length is ,k+1’*).*

The first relation will be:

$$k = C(2) = 2k \qquad (1)$$

where ,k+1’ is the length of ,n’ which is write in 2-base system.

We also have $n = 2^k + e(k-1) \cdot 2^{k-1} + \ldots + e(1) \cdot 2 + e(0)$ where it has been assumed that $e(k) = 1$ without restricting the generality.

Then n = $2^k \Leftrightarrow \lg_2 n = k \Rightarrow$

$$k = [\lg_2 n] \qquad (2)$$

which will be the second relation with ’[,]’ –whole part.

From (1) and (2) results $C(2) = 2k = 2[\lg_2 n]$ so the3 rdrelation:

$$C(2) = 2[\lg_2 n] \tag{3}$$

From the previous relation we will also have $n = 2^k + e(k-1)2^{k-1} + \ldots + e(1)\cdot 2 + e(0)$ and it results that $n \le 2^k + 2^{k+1} + \ldots + 2 + 1 = \frac{2^{k+1}-1}{2-1} = 2^{k+1} - 1 < 2^{k+1}$; *and we have* $n \le 2^{k+1}$.

So $[\lg_2 n] < k+1$ equivalent with the 4th relation:

$$[\lg_2 n] \le k\ . \tag{4}$$

From (1) and (4) results $[\lg_2 n] \le k \le C(2)$ therefore the 5th relation:

$$[\lg_2 n] = C(2) \tag{5}$$

From (6) and (7): $[\lg_2 n] = C(2) = 2[\lg_2 n]$ and from here it is obvious that $(\forall)\ n \in R\ (\exists)\ a_n \in R$ such that $C(2) = a_n \cdot [\lg_2 n]$; $a_n \in [1,2]$ and therefore:

$C(2) = O([\lg_2 n])$.

Proposition 3: The 2nd algorithm is one of the fastest way of finding x^n what it means: $C(g) \ge [\lg_2 n]$ where cost $C(g)$ is of the general algorithm (g) .

Demonstration: Every algorithm (it will be named general algorithm (g) of cost- *C(g))* must follow the steps.

The algorithm (g):

Step 1 : $y_1 := a_1 b_1$ where $a_1, b_1 \in \{1, x\}$ ……

Step k : $y_k := a_k b_k$ where a_k, b_k have been previously found meaning $a_k, b_k \in \{1, x, y_1, y_2, \ldots, y_{k-1}\}$

Step 'm': (the final step) $y_m := a_m b_m$; $a_m, b_m \in \{1, x, y_1, y_2, \ldots, y_{m-1}\}$

where $x^n = y_m$: which means that we have reached the result, the cost of the algorithm is *C(g)= m*.

Our 2nd algorithm also belongs to this class.

In the previous algorithm for $y_k := x^{e(k)}$ we have: $e(1) = \{0, 1, 2\}$.

In contrast with algorithm (2) where $e(1) \in \{0, 1\}$.

Then for general algorithm :

e(k) = e(i) + e(j) where 0 < i, j < k for k = 2 and e(m) = n .

Therefore $e(1) = 2$; $e(2) = e(1) + e(1) = 2^2$ and through induction results e(k) = 2^k; $(\forall)$k = 2.

In the end, $e(m) = n$ so, $e(m) = n = 2^m$ meaning $[\lg_2 n] = m \Rightarrow [\lg_2 n] = m = C(g)$

If we consider the cost classes according to a being $O[\, a\,]$ we will have using the 2nd algorithm an algorithm 'of the fastest class' because

$[\lg_2 n] = C(2) = 2\,[\lg_2 n]$ and $[\lg_2 n] = C(g)$.

2.Estimates of the costs of the algorithms used for conversions in p-base

Definition.3.

$$\boldsymbol{N}_p = \left\{ a \,/\, a = \sum_{i=0}^{n} a_i \cdot p^i, a_i, n \in \mathbf{N}; 0 \le a_i \le p-1 \right\}$$

are the p-adic numbers from ***N***

$$\boldsymbol{Z}_p = \left\{ a \,/\, a = \sum_{i=0}^{\infty} a_i \cdot p^i, a_i \in \mathbf{N}; 0 \le a_i \le p-1 \right\}$$

are the p-adic numbers from ***Z*** and we can define these numbers as the inverses of naturals numbers written in p-base system.

$$\boldsymbol{Q}_p = \left\{ a \,/\, a = \sum_{i=k}^{\infty} a_i \cdot p^i, k \in \mathbf{Z}, a_i \in \mathbf{N}; 0 \le a_i \le p_i - 1 \right\}$$

are the p-adic numbers from ***Q.***

Proposition 4.: The cost of writing an natural number '*x*' from the p-base system into 10-base system is $C(x) = O[\lg_2 (n!)]$

The natural numbers written in the p-base system can also be written ***Np*** and they are called p-adic numbers from ***N***.

Proof:

Be it $x \in \boldsymbol{N} \Rightarrow x_p = (a_n, a_{n-1} \ldots a_2, a_1, a_0)$ is the writing in a p-base system for x,

where $a_n \neq 0$ $a_i \in$ {0,1,….p-1}, for $i \in$ {0,1,2,…n-1} meaning:

$$x = a_{n-1} p^n + a_{n-1} p^{n-1} + \ldots + a_2 p^2 + a_1 p^1 + a_0$$

We consider the writing cost of a p-adic number represented by the number of multiplications and then:

$$C(x) = O[\lg_2 n] + O[\lg_2 (n-1)] + \ldots + O[\lg_2 2] + O[\lg_2 1]$$

$$C(x) = C_n \lg_2 n + C_{n-1} \lg_2 (n-1) + \ldots + C_1 \lg_2 1$$

where $C_n \in \mathbf{R}$ conveniently chosen.

Be it $\min\limits_{i \in \{1,2,\ldots,n]} c_i = c;$ $\max\limits_{i \in \{1,2,\ldots,n]} c_i = C$ then

$$c \cdot \sum_{i=1}^{n} \lg_2 i \leq C(x) \leq C \sum_{i=1}^{n} \lg_2 i \Leftrightarrow c \cdot \lg_2 (n!) \leq C(x) \leq C \cdot \lg_2 (n!) \Leftrightarrow$$

$$\mathrm{C(x) = O}\ [\lg_2 (n!)].$$

Obs. 1.: The number from $\boldsymbol{Z}$; $x \in \boldsymbol{Z}$; $x < 0$ seen as reverses to those from $\boldsymbol{N}$ in p-adic have a similar notation with the previous one; so: $x \in \boldsymbol{Z}$,$C(x)$ = $O\ [\lg_2 (n!)]$.

Obs.2.: The number from $\boldsymbol{Q}$ that have a 'p-adic' finite representation :

$$x = \overline{(a_n a_{n-1} \ldots a_1 a_0, a_{-1} \ldots a_{-m})} \text{ for } x = \sum_{i=-m}^{n} a_i \cdot p^{+i}; \quad m \in \boldsymbol{N},\ 0 = a_i < \mathrm{p}$$

are called p-adic number from $\mathbf{Q}$ into a finite representation.

Proposition 5.: The cost $C(x)$, $x \in \boldsymbol{Q}$ into a finite representation: $x = \sum\limits_{i=-m}^{n} a_i \cdot p^{+i}$; when 'conversing' in an abstract mode in the base-10 system with the previous relation , of a rational number ,is $C(x) = O[\lg_2 (m+n)!]$.

Proof:

From x$\in \mathbf{Q}$;

$$x = \sum_{i=-m}^{n} a_i p^i = a_n p^n + a_{n-1} p^{n-1} + \ldots + a_1 p^1 + a_0 + \frac{a_{-1}}{p} + \frac{a_{-2}}{p^2} + \ldots + \frac{a_{-m}}{p^m} \Rightarrow$$

$$x = \frac{a_n p^{m+n} + a_{n-1} p^{m+n-1} + \ldots + a_1 p^{m+1} + a_0 p^m + a_{-1} p^{m-1} + a_{-2} p^{m-2} + ..}{p^m}$$

$$\frac{.........+ a_{-m+1} p + a_{-m}}{p^m}$$

So x$\in$ **Q** and C(p ^{i}) = $O[\lg_2[i]] = C(a_n \cdot p^i)$, $i \in \{1,2...,m+p\}$ and

$O[\lg_2(m)]$ is the cost for the denominator p^m ,

$C(p^m) = O[\lg_2[m]] = C(a_n \cdot p^m)$

has been calculated.

So x$\in$**Q**; $C(x) = O[\lg_2(m+n)] + \ldots + O[\lg_2 1]$

$\Rightarrow C(x) = \lg_2[(m+n)!]$.

Obs.3.: The „costs" for a multiplication and for a divisation of two reals numbers with ‚n' bites , we can see in [3].

Conclusions

It is obvious that we must look for a less' expensive' method for mathematical calculations. As a result, less resources and computer operations will be used.

R E F E R E N C E S

1. *Gouvea, F.Q.* - *P*- adic Numbers .An introduction Second Edition..New York- Ed.:Springer-Verlag 1997
2. *Knuth, D.* – The Art of Programing vl.2 second edition - Ed:Addison Wesley 1981
3. *Mignotte, M*. - Introducere in algebra computationala si programare liniara - Editura Universitatii din Bucuresti 2000