CREATIVE MATH.
14 (2005), 95 - 99

Optimal values in structures with pseudo-random generated elements

CAMELIA-MIHAELA PINTEA

ABSTRACT. We propose structures with recurrent and random values. These values are called pseudo-random values. The goal is to find the optimal values for the pseudo-random values of dynamic lists, character strings and numerical sequences.

1. INTRODUCTION

In this paper we intend to find optimal values of structures whose values are generated using pseudo-random method. The mechanism is based on a congruential method.

The pseudo-random method is applied for dynamic lists, character strings and numerical sequences. Optimal values of each structure are determined. The main procedures and functions are described in Sections 3 and 4. The tests results are shown in Table 1, from Section 5.

With minimal modifications, the technique can be applied for sorting structures with pseudo-random values or other problems involving optimal values for pseudo-random values.

2. THEORETICAL SUPPORT

The proprieties and the main theorem are based on [**?**].

Definition 2.1. The pseudo-random generated values are values taken from a random sequence of numbers well determined from the predecessor by a given rule.

Remark 2.1. Any pseudo-random generated sequence is periodic.

Remark 2.2. The quality of a sequence is given by the length of its period.

Theorem 2.1. *The congruential sequence* $a[1]$, $a[2]$, ... *has pseudo-random values* $a[1]$, $a[2]$, ... *where* $a[k]$ *value is a recurrent one, described by:*

$$a[k] := (m * a[k-1] + i) \mod modulo$$

This sequence has a maximal modulo length if and only if:

1) *modulo and* i*, the increment, are integers and relative prime numbers.*
2) $b = m - 1$ *is a multiple for any prime number which divides modulo*
3) b *is a multiple of* 4 *if modulo is a multiple of* 4.

Received: 17.12.2004. In revised form: 03.03.2005.
2000 *Mathematics Subject Classification.* 11K45, 68W01.
Key words and phrases. *Pseudo-random, dynamic structures, static structures.*

3. Algorithm description

The algorithm is using *Theorem 2.1.* The random mechanism that generates numeric sequences is based on a congruential method.

The congruential method is stated by a numeric pseudo-random sequence $a[1]$, $a[2]$,... where $a[k]$ has a recurrent definition.

$$a[k] := (m * a[k-1] + i) \mod modulo$$

The main functions of the algorithm are shown below. The pseudo-random sequences are generated using *Pseudo* function. The *Max2* function is the classical function that returns the maximal value of two elements.

```
function pseudo
const
   modulo = 128; m = 33; i = 17;
begin
   r := (m * r + i)   mod  modulo;
   pseudo := r;
end;

function max2(a,b:integer):integer;
begin
   if (a >= b) then max2 := a else max2 := b;
end;
```

4. Structures with pseudo-random values

We will apply the method described above to obtain values for dynamic lists, character strings and numerical sequences. For each structure it is determined the maximal value.

The procedures used for these structures are described in this section. The functions and procedure are in Pascal language, but could be considered as an Pseudo-code and be written in any other programming language.

Numerical sequences

The elements of the numerical sequence are generated using the algorithm described in the previous section. For every i, the numerical sequence a is generated using the $Pseudo$ function:

$$a[i] := Pseudo(random)$$

The recurrent procedure for the maximal value of a numerical sequence is described below.

```
Procedure MaxN (var x:integer;var k:integer;var maxi:integer);
begin
   if k <= n then
   begin
      maxi := Max2(x, maxi);
      k := k + 1;
      MaxN(a[k], k, maxi);
   end;
end;
```

The maximal value max is returned by $MaxN$ procedure with the parameters: $MaxN(a[1], i, max)$.

Character Strings

To obtain the character strings it is used $Pseudo$ function. Let denote s the character string. It is used, also, a function, $chr()$, that returns the character of an $ASCII$ code.

$$s := s + chr(Pseudo(random))$$

In the following, it is shown the recurrent procedure used to obtain the maximal value for the character string s. The function $ord()$ returns the $ASCII$ code of a character.

```
Procedure MaxChar(var x:char;var k:integer;var maxi:char);
begin
   if k <= n then
   begin
      maxi := chr(Max2(ord(x), ord(maxi)));
      k := k + 1;
      MaxChar(s[k], k, maxi);
   end;
end;
```

The maximal value of s is max from $MaxChar(s[1], i, max)$.

Dynamic lists

A dynamic list, L, with the pseudo-random elements is generated using an auxiliary p variable:

$$new(p);\ p^\wedge.info := Pseudo(random);\ p^\wedge.next := L;\ L := p;$$

where $info$ is the information from a node and $next$ is the link with the following node of the list.

After we generated the list, we will find the maximal element, using the $MaximN$ procedure.

Procedure MaximN (var p:Plista;var X:integer;var maxi:integer);
begin
if $p^\wedge.next <> NIL$ then
begin
$maxi := Max2(p^\wedge.info, maxi)$;
$p := p^\wedge.next$;
$MaximN(p, p^\wedge.info, maxi)$;
end;
end;

The maximal value for the dynamic list is $max := p^\wedge.info$ where $MaximN(p, p^\wedge.info, max)$.

5. Tests and results

We tested the algorithm for five elements. The pseudo-random values and the optimal values are shown in Table 1.

TABLE 1

			Max		Min	
No.	Num.Sequence	Char.Sequence	Num.	Char.	Num.	Char.
1	$50, 3, 116, 5, 54$	$2, \heartsuit, t, \clubsuit, 6$	116	t	3	$\heartsuit$
2	$83, 68, 85, 6, 87$	$S, D, U, \spadesuit, W$	87	W	6	$\spadesuit$
3	$116, 5, 54, 7, 120$	$t, \clubsuit, 6, BEL, x$	120	x	5	$\clubsuit$
4	$21, 70, 23, 8, 25$	$\S, F, \ddagger, BS, \downarrow$	70	F	8	BS
5	$54, 7, 120, 9, 58$	$6, BEL, x, o, :$	120	x	7	BEL
6	$87, 72, 89, 10, 91$	$W, H, Y, LF, [$	89	Y	10	LF
7	$120, 9, 58, 11, 124$	$x, o, :, \pm, \vert$	124	$\vert$	9	o
8	$25, 74, 27, 12, 29$	$\downarrow, J, \leftarrow, \mp, \leftrightarrow$	74	J	12	$\mp$
9	$58, 11, 124, 13, 62$	$:, \pm, \vert, CR, >$	124	$\vert$	11	$\pm$
10	$91, 76, 93, 14, 95$	$[, L,], \sharp, -$	95	$-$	14	$\sharp$

The ten tests show the pseudo-random values for the mentioned structures. The maximal and minimal values are for each sequence, local optimal values. The global values from Table 1 are: 124 (|) maximal value and 3($\heartsuit$) minimal value.

For static sequences and dynamic lists the values are the same. The character strings are the *ASCII* codes of the numerical sequences.

6. Conclusions

Using a congruential method, we have generated structures with pseudo-random values. Dynamic lists, character strings and numeric sequences were obtained with this method. For each structure, using recurrent procedures, were determined maximal values.

The maximal values found are in the same domain as *modulo* variable. That is why the maximal value is between 0 and $modulo - 1$. It is obvious that the global maximal value is $modulo - 1$. The minimal values are obtained in the same way. The global minimal value is 0.

The mentioned optimal values could be also used in other problems such as sorting sequences with pseudo-random values.

These applications are interesting from mathematical point of view and they are useful to consolidate the elementary programming techniques.

References

[1] Cormen T.H., Leiserson C.E., Rivest R.L., *Introducere în Algoritmi*, Computer Libris Agora, 2000

[2] Ivasc C., Pruna M., *Tehnici de programare - Aplicaţii*, Editura Petrion, 1999

[3] Lucanu D., *Proiectarea Algoritmilor*, Editura Universităţii "Al.I. Cuza", Iaşi, 1993

[4] Lucanu D., *Bazele proiectării programelor şi algoritmilor*, Univ. "Al.I. Cuza", Iaşi, 1996

[5] Livovshi L., Georgescu H., *Sinteza şi analiza algoritmilor*, Editura Ştiinţifică şi Enciclopedică, 1986

[6] Sorin T., *Tehnici de programare*, Editura Teora, 1994

Colegiul Naţional "George Coşbuc"
Avram Iancu 70, 400083 Cluj Napoca, Romania
E-mail address: cami_mihaela@yahoo.com