Lista 2 – Modelagem e Simulação

Grupo: Lucas Moreira Magalhães
Marcus Adriano Pereira
Vitor Hugo Honorato Tiago

Questão 2.1.1 -
For the tiny Lehmer generator defined by g(x) = ax mod 127, find all the full-period multipliers.
(a) How many are there?
(b) What is the smallest multiplier?

Resposta -
(a) Como 127 é um número primo, podemos encontrar o número de full-period multipliers através do teorema 2.1.3. Assim sendo, (127-1) pode ser fatorado em $2^1.3^2.7^1$, com isso o cálculo do número de full-period multipliers fica:

$$\frac{(2-1).(3-1).(7-1)}{2.3.7}.(127-1) = 36$$

Logo, temos 36 full-period multipliers.

(b) Código C em anexo. O menor multiplicador é 3.

Questão 2.1.6 -
In ANSI C an int is guaranteed to hold all integer values between −(2^15 − 1) and 2^15 − 1 inclusive.
(a) What is the largest prime modulus in this range?
(b) How many corresponding full-period multipliers are there and what is the smallest one?

Resposta -
(a) 32749 é o maior número primo nesse intervalo.

(b) Para descobrir quantos full-period multipliers existem iremos utilizar novamente o teorema 2.1.3. Assim sendo, fatoramos (32749 – 1), e obtemos $2^2.3^1.2729^1$, com isso o cálculo do numero de full-period multipliers fica:

$$\frac{(2-1).(3-1).(2729-1)}{2.3.2729}.(32749-1) = 10912$$

Concluímos então que existem 10912 full-period multipliers para m = 32749. Através de um algoritmo (Código C em anexo) conseguimos encontrar o menor full-period multiplier que é 2.

Questão 2.1.8 -
(a) Evaluate 7^i mod 13 and 11^i mod 13 for i = 1, 5, 7, 11.
(b) How does this relate to Example 2.1.5?

Respostas -
(a) Avaliando $7^i mod\ 13$ e $11^i mod\ 13$ para i = 1, 5, 7, 11
$7^1 mod\ 13 = 7$, $7^5 mod\ 13 = 11$, $7^7 mod\ 13 = 6$, $7^{11} mod\ 13 = 2$.

$11^1 mod\ 13 = 11$, $11^5 mod\ 13 = 7$, $11^7 mod\ 13 = 2$, $11^{11} mod\ 13 = 6$.

(b) O Exemplo 2.1.5 é relacionado ao teorema 2.1.4, que indica como podemos obter todos os full-period multipliers de um modulo primo m. Para tal, é feito:

$$a^i mod\ m \in X_m \qquad i = 1, 2, 3, ..., m - 1.$$

onde todos os *i* são primos relativos a m-1.
Portanto no caso do exemplo acima, considerando m = 13, temos que para (m-1) os índices variados de i serão 1, 5, 7, 11. O exemplo 2.1.5 e a análise feitas acima na letra (a) se complementam para exemplificar o funcionamento do teorema 2.1.4. Podemos analisar que no exemplo foram obtidos os números 6, 2 , 7 e 11, utilizando a = 6 e 2. Nos cálculos feitos na letra (a), obtivemos os mesmos números 6, 2, 7 e 11, utilizando a = 7 e 11. Portanto, concluímos que, independente de qual full-period multiplier inicial for utilizado, as equações do teorema 2.1.4 levam aos outros full-period multiplier.

Questão 2.1.9 -
(a) Verify that the list of five full-period multipliers in Example 2.1.6 is correct.
(b) What are the next five elements in this list?
Respostas -
(a) Os 5 números apresentados no Exemplo 2.1.6 estão corretos, portanto, são full-period multipliers. Código C que faz essa verificação em anexo.

```
lucas@iceberg:~/Área de Trabalho$ gcc 2_1_9a.c -o out.o
lucas@iceberg:~/Área de Trabalho$ ./out.o
7 is full-period multiplier relative to 2147483647
16807 is full-period multiplier relative to 2147483647
252246292 is full-period multiplier relative to 2147483647
52958638 is full-period multiplier relative to 2147483647
447489615 is full-period multiplier relative to 2147483647
```

(b) Utilizando o teorema 2.1.4 podemos calcular os próximos cinco full-period multipliers através dos índices *i*, primos relativos a *m*. Os próximos índices serão: 23, 29, 37, 41, 43. Logo, os próximos elementos da lista serão:

$$7^{23} mod\ 2147483647 = 680742115$$
$$7^{25} mod\ 2147483647 = 1144108930$$

$$7^{29} mod\ 2147483647 = 373956417$$
$$7^{37} mod\ 2147483647 = 655382362$$
$$7^{41} mod\ 2147483647 = 1615021558$$

```
gcc 2 1 9b.c -o bin/219b.out
marcus@mars:~/Documents/UFU/2018-1/MS/2$ ./bin/219b.out
7 is full-period multiplier relative to 2147483647
16807 is full-period multiplier relative to 2147483647
252246292 is full-period multiplier relative to 2147483647
52958638 is full-period multiplier relative to 2147483647
447489615 is full-period multiplier relative to 2147483647
680742115 is full-period multiplier relative to 2147483647
1144108930 is full-period multiplier relative to 2147483647
373956417 is full-period multiplier relative to 2147483647
655382362 is full-period multiplier relative to 2147483647
1615021558 is full-period multiplier relative to 2147483647
```

Questão 2.1.11 -
For the first few prime moduli, this table lists the number of full-period multipliers and the smallest full-period multiplier. Add the next 10 rows to this table.

prime modulus m	number of full-period multipliers	smallest full-period multiplier a
2	1	1
3	1	2
5	2	2
7	2	3
11	4	2
13	4	2

Resposta -
Próximas 10 linhas da tabela:

17	8	3
19	6	2
23	10	5
29	12	2
31	8	3
37	12	2
41	16	6
43	12	3
47	22	5
53	24	2

Fórmula utilizada para descobrir o número de full-period multipliers, através do teorema 2.1.3.

$$\frac{(p_1 - 1)(p_2 - 1) \ldots (p_r - 1)}{p_1 p_2 \ldots p_r} (m - 1).$$

Código C em anexo para encontrar o menor full-period multiplier.

Questão 2.2.9-

You have been hired as a consultant by XYZ Inc to assess the market potential of a relatively inexpensive hardware random number generator they may develop for high-speed scientific computing applications. List all the technical reasons you can think of to convince them this is a bad idea.

Respostas –

Um gerador de números aleatórios deve seguir alguns princípios que não podem ser alcançados através da utilização de um gerador por hardware. Controlabilidade e portabilidade são as principais características que não podem ser implementadas em um gerador de números aleatórios através de hardware. A controlabilidade indica a capacidade de reproduzir uma mesma saída se desejado, enquanto a portabilidade é a capacidade de produzir a mesma saída para diversos computadores em um sistema, como o gerador de números aleatórios por hardware utiliza informações únicas e específicas em cada momento, logo não é possível reproduzir novamente uma sequencia, mesmo se desejado, e nem gerar a mesma sequência aleatória em diferentes computadores. Considerando isso, é muito mais viável e produtivo aderir um gerador aleatório através de software do que um gerador aleatório que utiliza-se de hardware.

Questão 2.2.11 -

Let m be the largest prime modulus less than or equal to 2^15 − 1 (see Exercise 2.1.6).
(a) Compute all the corresponding modulus-compatible full-period multipliers.
(b) Comment on how this result relates to random number generation on systems that support 16-bit integer arithmetic only.

Respostas -

(a) Código C em anexo. O número de full-period multipliers é 10912.

(b) Considerando um sistema onde a representação do inteiro é dada em 16 bits e que o maior primo no intervalor [1, 2^15 – 1] é 32749 temos um total de $\frac{10912}{32749}$ = 33% full-period multipliers (um número maior que a representação de 32 bits que é 25%). Em contrapartida, se m = 32749, o conjunto de Xn é relativamente menor do que em um sistema que usa inteiros de 32 bits, onde m = 2^{31} - 1.