UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO

Trabalho de Modelagem e Simulação
Prática 02

Alunos: Aline de Souza Lima Abreu
Miguel Henrique de Brito Pereira
Tarcísio Magno de Almeida Filho
Vinícius Gonzaga Rocha

Uberlândia
2017

Sumário

Lista de Figuras

Exercício 2.1.1

2.1.1) For the tiny Lehmer generator defined by g(x) = ax mod 127, find all the full-period multipliers.

Com m=127 temos m-1 = 126, fazendo a v em números primos de 126, temos: m-1 = 2 * 3^2 * 7

Com base na implementação simples do Algoritmo 2.1.1 identificamos quais são os multiplicadores full-período e podemos confirmar o cálculo das quantidades, de acordo com a letra "a"deste exercício.

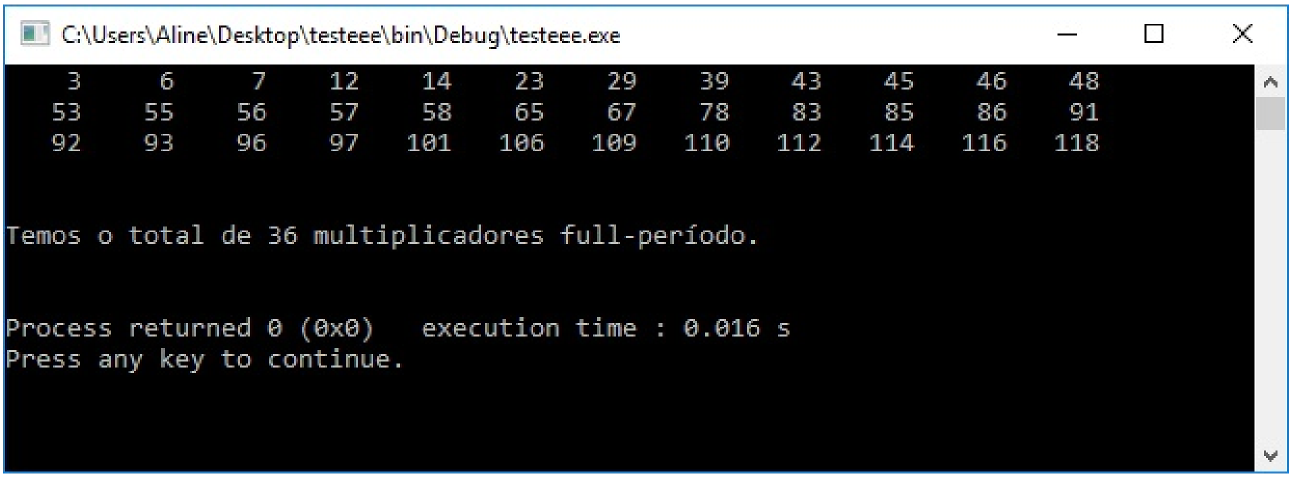

Figura 1: Output alteração Ex.2.1.1

a) How many are there?

$\frac{(2-1)*(3-1)*(7-1)}{2*3*7}*(2*3^2*7) = 36$

b) What is the smallest multiplier?

Analisando o output do algoritmo 2.1.1, temos que o menor multiplicador full-período é o número 3.

Exercício 2.1.6

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?

No intervalo dado temos o total de 36282 números primos, o maior deles é 32749.

b) How many corresponding full-period multipliers are there and what is the smallest one?

Utilizando o algoritmo 2.1.1, sabemos que temos o total de 10.912 multiplicadores full-período, o menor deles é o número 2. A escolha do módulo foi dada pegando o maior número primo possível do intervalo (m=32749), a semente inicial foi 1.

Cálculo da quantidade de multiplicadores full-período:

$\frac{(2-1)*(3-1)*(2729-1)}{2*3*2729}*(2^2*3*2729) = 10912$

Exercício 2.1.8

2.1.8) a) Evaluate 7^i mod 13 and 11^i mod 13 for i = 1, 5, 7, 11.

7^1 mod 13 = 7
7^5 mod 13 = 11
7^7 mod 13 = 6
$7^1 1$ mod 13 = 2

11^1 mod 13 = 11
11^5 mod 13 = 7
11^7 mod 13 = 2
$11^1 1$ mod 13 = 6

b) How does this relate to Example 2.1.5?

De acordo com o exemplo 2.1.5 sabemos que existem 4 inteiros entre 1 e 12 que são primos relativos à 12 (1, 5, 7 e 11), com o algoritmo 2.1.1 sabemos que os multiplicadores full-período relativos à 13 são 2, 6, 7 e 11.

No teorema 2.1.4 temos que se o "a"é qualquer multiplicador full-período relativo ao módulo primo (m), então a^i mod m irá sempre pertencer ao conjunto Xm, desde que i seja primo relativo à m-1.

Com isto temos que os resultados para 7^i mod 13 e 11^i mod 13 tendo os valores de i = 1, 5, 7, 11, serão os próprios valores dos multiplicadores full-período, assim como acontece para 6^i mod 13 e 2^i mod 13 no exemplo 2.1.5.

Exercício 2.1.9

2.1.9) a) Verify that the list of five full-period multipliers in Example 2.1.6 is correct.

Sim, de acordo com o teorema 2.1.4, está correto pois os valores de i (1, 5, 13, 17 e 19) são primos relativos à m-1=2147483646.

b) What are the next five elements in this list?

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

Exercício 2.1.11

2.1.1) 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

Valor de m	Full period multipliers	Menor full period multiplier
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

Figura 2: Solução Ex.2.1.11

Exercício 2.2.9

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.

Como dito na seção 2.1.1 do livro, apenas geradores do tipo “white noise” são realmente aceitos. E somente softwares geradores conseguiram alcançar desejavelmente os critérios de numeros randômicos que são:

- *Random* - capaz de produzir saidas que passam em todos os testes estatísticos de aleatoriedade;
- *Controllable* - capaz de ser controlavel;
- *Portable* - capaz de produzir as mesmas saídas em vários sistemas;
- *Efficient* - rápido e com requisitos mínimos;
- *Documented* - teoricamente analisado e amplamente testado.

Exercício 2.2.11

2.2.11) Let m be the largest prime modulus less than or equal to 2^{15} - 1 (see Exercise 2.1.6).

O maior primo menor ou igual a 32767 (2^{15} - 1) é 32749.

a) Compute all the corresponding modulus-compatible full-period multipliers.

Um total de 10192 full-period Multipliers foi calculado e segue em arquivo separado.

b) Comment on how this result relates to random number generation on systems that support 16-bit integer arithmetic only.

O resultado seria obtido em uma máquina de representação 16 bits, onde o máximo inteiro unsigned alcançado é 65535 (2^{16} - 1). Porém, este seria o máximo m suportado (32749), uma vez que o próximo primo é 32771 e sendo 2 o menor FPM para 32771 e também o FPM escolhido para a, a multiplicação $a*x$ excederia 65535 3 vezes (em x = 32768, 32769 e 32770).
Caso não fosse utilizado unsigned o resultado seria ainda menor, uma vez que o máximo inteiro alcançado seria 32767 e o overflow ocorreria várias vezes.
Portanto, em máquinas de 16 bits o máximo m que pode ser utilizado para o algoritmo 2.1.1 é 32749, e $Xm = 1, \ldots, 32748$.
Entretanto, o algoritmo 2.2.1 garante que nenhum produto $a*x$ seja maior que $m\text{-}1$, uma vez que a operação mod é realizada antes da multiplicação. Neste caso, até $m = 65563$ e $Xm = 1, \ldots, 65562$ podem ser gerados.

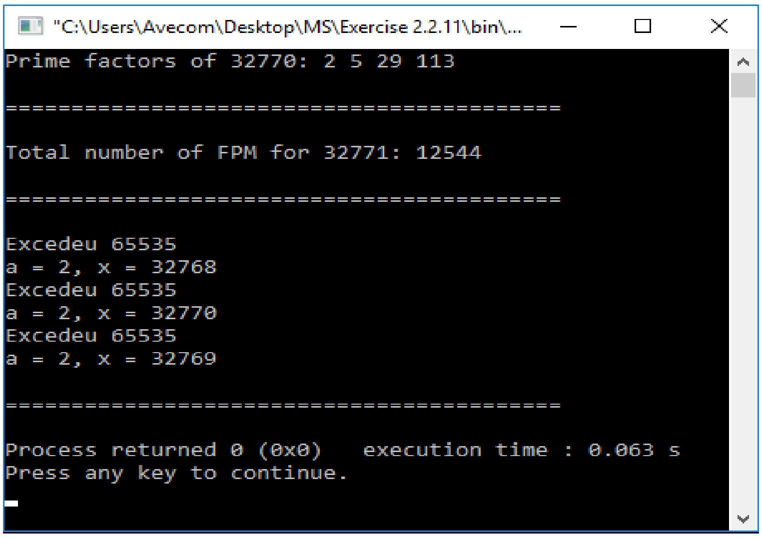

Figura 3: Output alteração algoritmo 2.1.1, a=2, m=32771

Exercício 2.2.15

2.2.15) Determine whether the multipliers associated with m = 2^{31} - 1 given by Fishman (2001): a = 630 360 016, a = 742 938 285 a = 950 706 376, a = 1 226 874 159, a = 62 089 911, and a = 1 343 714 438 are modulus-compatible.

A definição de modulus-compatible fornecida pelo livro é a de que o resto da divisão de m por a deve ser estritamente menor que o cosciente.

Definition 2.2.1 The multiplier a is modulus-compatible with the prime modulus m if and only if the remainder $r = m \bmod a$ is less than the quotient $q = \lfloor m/a \rfloor$.

Multipliers	**MC?**	**r**	**q**
630360016	Não	256403599	3
742938285	Não	661607077	2
950706376	Não	246070895	2
1226874159	Não	920609488	1
62089911	Não	36426673	34
1343714438	Não	803769209	1

Tabela 1: Modulus compatibility dos valores fornecidos por Fishman (2001) para m = $2^{31} - 1$

Exercício 2.3.6

2.3.6) According to slides number seven and eight from section 2.3, example 2.3.6, construct a graph similar to slide eight but Pr(X=9).

Para construir o gráfico foi necessário realizar uma simulação de Monte Carlo para o arremesso de 3 dados não viciados. Foram utilizadas as função Random(), PutSeed() e Equilikely(), todas estas presentes neste link.
Uma vez que estas funções estavam disponíveis, foi implementada a função Games(), que recebia N (número de jogadas) e win (soma que signifacaria uma vitoria) e retornava a porcentagem de vitórias dividido pelo número de jogadas. Na função main() foram setadas 3 sementes diferentes e um loop for que inciava em 20 (mínimo de jogadas) e se incrementava de 20 em 20 até 1000 (máximo de jogadas). Todos os resultados foram salvos em arquivos csv.
Em seguida, os arquivos foram abertos pelo script em python e plotados no gráfico com a biblioteca matplotlib.
Verifica-se que lentamente que os resultados obtidos vão de encontro a probilidade axiomática, que é de $25/216 = 0.116$, conforme o número de jogadas cresce.

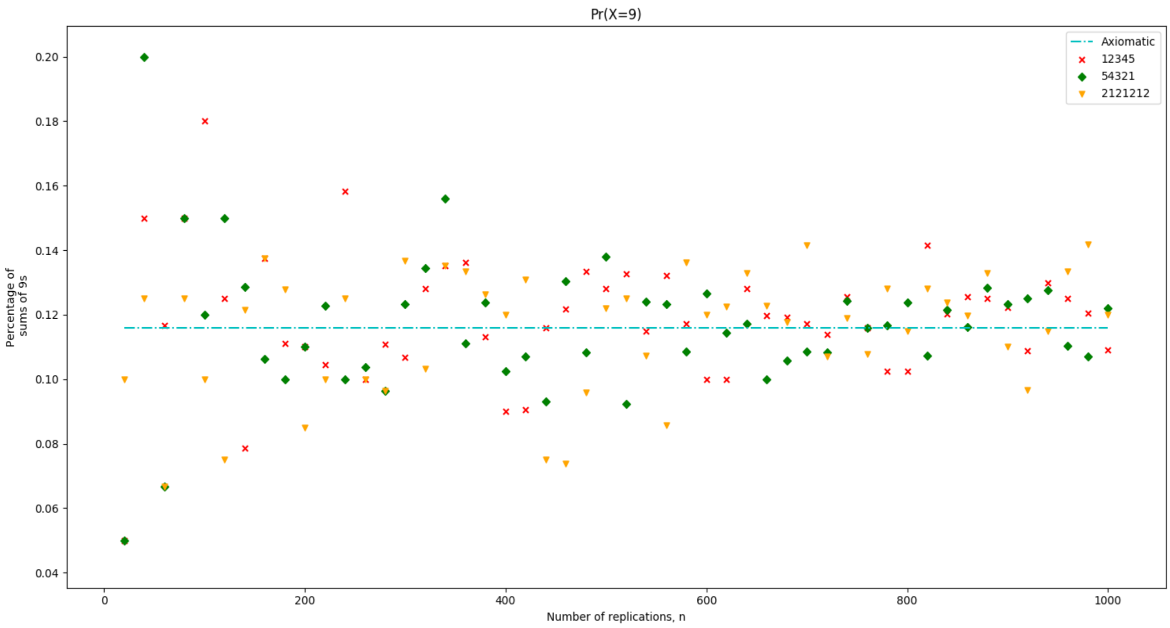

Figura 4: Monte Carlo para 3 dados e soma = 9