INSTITUTO TECNOLÓGICO DE CIUDAD MADERO.



UNIDAD 2

PRACTICA 6

ALUMNA: CAVAZOS ARGOT ANA VICTORIA

N° CONTROL: 15071292

PROFESOR: DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN

MATERIA: PROGRAMACIÓN PARALELA

FECHA DE ENTREGA: 15 DE OCTUBRE 2018

Índice:

[Ejercicio 2: 3](#_Toc524905251)

[Introducción: 3](#_Toc524905252)

[Marco teórico: 3](#_Toc524905253)

[Media: 3](#_Toc524905254)

[Moda: 3](#_Toc524905255)

[Desviación estándar: 3](#_Toc524905256)

[Varianza: 3](#_Toc524905257)

[Metodología: 4](#_Toc524905258)

[Experimentación y resultados: 5](#_Toc524905259)

[Conclusiones: 5](#_Toc524905260)

[Bibliografía: 5](#_Toc524905261)

Ejercicio:

Introducción:

Hacer un programa que reciba m números de entrada aleatorios e imprima m, la secuencia de números generados y la cantidad de números generados.

**The 3n + 1 Problem**

Consider the following algorithm to generate a sequence of numbers. Start with an integer n. If n is even, divide by 2. If n is odd, multiply by 3 and add 1. Repeat this process with the new value of n, terminating when n = 1. For example, the following sequence of numbers will be generated for n = 22:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured (but not yet proven) that this algorithm will terminate at n = 1 for every integer n. Still, the conjecture holds for all integers up to at least 1, 000, 000. For an input n, the cycle-length of n is the number of numbers generated up to and including the 1. In the example above, the cycle length of 22 is 16. Given any two numbers i and j, you are to determine the maximum cycle length over all numbers between i and j, including both endpoints.

Marco teórico:

Conjetura de Collatz o 3n+1:

En 1937 el matemático alemán Lothar Collatz propuso un problema que hasta la fecha no se ha podido demostrar. Lo bonito y sorprendente de este problema es que, a partir de cualquier número natural, siempre se obtenemos la unidad.

Elige cualquier número natural (n) y realiza los siguientes cálculos:

Si n es par divide entre 2 (Es decir n/2)

Si n es impar multiplica por 3 y suma 1 al resultado (Es decir 3n+1)

Con el número que hayas obtenido tienes que repetir el proceso. Así sucesivamente hasta que el numero sea 1. Sin embargo, la secuencia crece muy rápido. Para una número pequeño como el 27 se obtiene una secuencia de 111 pasos.

Metodología:

Conjunto de datos:   
**Números aleatorios generados:** 5,6,7,8,9,10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Resultados por numero:** | | | | | |
| **1** | **10** | **9** | **8** | **7** | **6** | **5** |
| **2** | 5 | 13 | 4 | 22 | 3 | 16 |
| **3** | 16 | 38 | 2 | 11 | 10 | 8 |
| **4** | 8 | 19 | 1 | 34 | 5 | 4 |
| **5** | 4 | 58 |  | 17 | 16 | 2 |
| **6** | 2 | 29 |  | 52 | 8 | 1 |
| **7** | 1 | 88 |  | 26 | 4 |  |
| **8** |  | 44 |  | 13 | 2 |  |
| **9** |  | 22 |  | 40 | 1 |  |
| **10** |  | 11 |  | 20 |  |  |
| **11** |  | 34 |  | 10 |  |  |
| **12** |  | 17 |  | 5 |  |  |
| **13** |  | 11 |  | 16 |  |  |
| **14** |  | 64 |  | 8 |  |  |
| **15** |  | 32 |  | 4 |  |  |
| **16** |  | 16 |  | 2 |  |  |
| **17** |  | 8 |  | 1 |  |  |
| **18** |  | 4 |  |  |  |  |
| **19** |  | 2 |  |  |  |  |
| **20** |  | 1 |  |  |  |  |

**Calculo paso por paso:**

**n:** 10 **iteraciones**: 7

1) n = 10 es par

Entonces:

n = (n/2)

2) n = 5 es impar

Entonces:

n = (n\*3) + 1

3) n = 16 es par

Entonces:

n = (n/2)

4) n = 8 es par

Entonces:

n = (n/2)

5) n = 4 es par

Entonces:

n = (n/2)

6) n = 2 es par

Entonces:

n = (n/2)

7) n = 1

Experimentación y resultados:

Información sobre el equipo:

**Modelo**: Dell OptiPlex 7010

**Procesador**: Intel(R) Core(TM) i5-3550 CPU @ 3.30GHz

**Memoria RAM**: 4.00 GB

**Tipo de sistema**: Sistema operativo de 64 bits

**Sistema operativo utilizado**: Windows 7 Ultimate Service Pack 1

Tabla de resultados:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Experimento | SEED | Cantidad de secuencias | Ultimo valor | Secuencia | Números generados | Tiempo secuencial  (segundos) | Hilos | Tiempo paralelo OMP  (segundos) |
| 1 | 45 | 10,000 | 768 | 768..1 | 16 | **50.773** | 2 | 52.291 |
| 8 | 38.717 |
| 10 | 46.032 |
| 20 | 41.559 |
| 2 | 45 | 20,000 | 10876 | 10876…1 | 69 | **98.4280** | **2** | 96.524 |
| **8** | 96.496 |
| **10** | 95.753 |
| **20** | 96.424 |
| 3 | 45 | 30,000 | 29738 | 29738… | 135 | **124.301** | 2 | 141.491 |
| 8 | 141.58 |
| 10 | 140.231 |
| 20 | 124.927 |
| 40 | 123.197 |
| 100 | 124.082 |
| 4 | 45 | 40,000 | 4798 | 4798..1 | 73 | **204.173** | **2** | 169.32 |
| **8** | 167.351 |
| **10** | 167.678 |
| **20** | 167.646 |
| 5 | 45 | 50,000 | 29693 | 29693…1 | 241 | **210.731** | **2** | 210.729 |
| **8** | 209.097 |
| **10** | 209.331 |
| **100** | 210.827 |

Conclusiones:

Durante la realización de esta práctica se optó por solo paralelizar el cálculo recursivo de la serie. Es posible paralelizar el proceso donde se obtiene el valor inicial de la serie, sin embargo esto provoca que los resultados varíen entre cada caso de prueba por lo que no puede ser comparado con el programa secuencial.

Con la paralelizacion de funciones aplicada se logró reducir el tiempo de ejecución en la mayoría de casos utilizando 8 o 10 hilos, sin embargo, en el caso 3 donde se utiliza 30,000 datos el programa paralelizado requiere de una gran cantidad de hilos para apenas alcanzar el tiempo de ejecución secuencial. No encuentro una explicación para esto ya que al aumentar la cantidad de datos utilizados la paralelizacion si mejora el tiempo de ejecución con menos hilos.

Bibliografía:

<https://soymatematicas.com/conjetura-matematica/>