INSTITUTO TECNOLÓGICO DE CIUDAD MADERO.



UNIDAD 1

PRACTICA 2

ALUMNA: CAVAZOS ARGOT ANA VICTORIA

N° CONTROL: 15071292

PROFESOR: DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN

MATERIA: PROGRAMACIÓN PARALELA

FECHA DE ENTREGA: 16 DE SEPTIEMBRE 2018

Índice:

[Ejercicio 1: 3](#_Toc524887500)

[Introducción: 3](#_Toc524887501)

[Marco teórico: 3](#_Toc524887502)

[Grafo: 3](#_Toc524887503)

[Clasificación de los grafos: 3](#_Toc524887504)

[Grafo completo: 3](#_Toc524887505)

[Grafo conexo: 3](#_Toc524887506)

[Grafo euleriano: 3](#_Toc524887507)

[Metodología: 4](#_Toc524887508)

[Conclusiones: 7](#_Toc524887509)

[Bibliografía: 7](#_Toc524887510)

[Ejercicio 2: 8](#_Toc524887511)

[Introducción: 8](#_Toc524887512)

[Marco teórico: 9](#_Toc524887513)

[Media: 9](#_Toc524887514)

[Moda: 9](#_Toc524887515)

[Desviación estándar: 9](#_Toc524887516)

[Varianza: 9](#_Toc524887517)

[Números pares e impares: 9](#_Toc524887518)

[Números primos: 9](#_Toc524887519)

[Test de primalidad: 10](#_Toc524887520)

[Números amigos: 10](#_Toc524887521)

[Metodología: 11](#_Toc524887522)

[Experimentación y resultados: 13](#_Toc524887523)

[Conclusiones: 14](#_Toc524887524)

[Bibliografía: 14](#_Toc524887525)

Ejercicio 1:

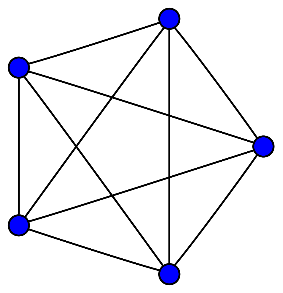
Introducción:

Diseñe un algoritmo o diagrama de flujo que identifique: a) si un grafo es completo b) si un grafo es conexo y c) tiene un ciclo Euleriano.

Marco teórico:

Grafo:

Un grafo consiste de un conjunto de vértices y otro conjunto de aristas que unen algunos de los vértices.

Clasificación de los grafos:

Grafo completo:

Un grafo es completo si existen aristas uniendo todos los pares posibles de vértices. Es decir, todo par de vértices (a,b) debe tener una arista e que los une.

Grafo conexo:

Un grafo es conexo si cada par de vértices está conectado por un camino; es decir, si para cualquier par de vértices (a, b), existe al menos un camino posible desde a hacia b.

Grafo euleriano (ciclo):

Un grafo (dígrafo) euleriano es aquel en que pueden recorrerse todas sus las aristas (arcos) de manera consecutiva y sin repetirlas.

Un grafo conexo y no dirigido se dice que es euleriano si cada vértice tiene un grado par o solo 2 vértices tienen grado impar.

Metodología:

Grafo: 1  
**Vectores:** 5

**Coordenadas:**

(1,2),(1,3),(1,4),(2,3),(2,4),(4,3),(4,5),(5,1)

**Tabla de datos:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nodos | Rutas | | | | | Total de rutas por nodo |
| **1** | **2** | **3** | **4** | **5** |
| **1** | 0 | 1 | 1 | 1 | 1 | 4 |
| **2** | 1 | 0 | 1 | 1 | 0 | 3 |
| **3** | 1 | 1 | 0 | 1 | 0 | 3 |
| **4** | 1 | 1 | 1 | 0 | 1 | 4 |
| **5** | 1 | 0 | 0 | 1 | 0 | 2 |

**a) ¿Es completo?**:

Se necesita que cada nodo posea n-1 rutas para el grafo completo: **n-1 rutas = 5 - 1 = 4**

|  |  |  |
| --- | --- | --- |
| Nodos | Total de rutas por nodo | ¿Total es igual que n-1? |
| **1** | 4 | **Si** |
| **2** | 3 | **No** |
| **3** | 3 | **No** |
| **4** | 4 | **Si** |
| **5** | 2 | **No** |

Ya que no todos los nodos cumplen con la condición es grafo **NO ES COMPLETO**.

**c) ¿Es Euleriano?**:

Se necesita que de cualquier nodo a se pueda llegar a un nodo b recorriendo todas sus aristas y sin repetirlas.

Si todos los nodos del grafo poseen grado par rutas es Euleriano o solo dos nodos poseen rutas impares.

|  |  |  |
| --- | --- | --- |
| Nodos | Total de rutas por nodo | Grado |
| **1** | 4 | Par |
| **2** | 3 | Impar |
| **3** | 3 | Impar |
| **4** | 4 | Par |
| **5** | 2 | Par |

Ya que solo dos nodos poseen grado impar el grafo **ES EULERIANO**.

**b) ¿Es conexo?**:

Se necesita que de cualquier nodo a se pueda llegar a un nodo b.

Si un grafo es **completo** significa que también es **conexo**.

|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | | ¿Existe una  ruta directa? | Ruta  posible |
| a | b |
| 1 | 2 | Si |  |
| 1 | 3 | Si |  |
| 1 | 4 | Si |  |
| 1 | 5 | Si |  |
| 2 | 1 | Si |  |
| 2 | 3 | Si |  |
| 2 | 4 | Si |  |
| 2 | 5 | No | 2,1,5 |
| 3 | 1 | Si |  |
| 3 | 2 | Si |  |
| 3 | 4 | Si |  |
| 3 | 5 | No | 3,1,5 |
| 4 | 1 | Si |  |
| 4 | 2 | Si |  |
| 4 | 3 | Si |  |
| 4 | 5 | Si |  |
| 5 | 1 | Si |  |
| 5 | 2 | No | 5,1,2 |
| 5 | 3 | No | 5,1,3 |
| 5 | 4 | Si |  |

Ya que es posible llegar de cualquier nodo a cualquier otro dentro del grafo, este **ES CONEXO**.

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 | Vértices | Resultados | | |  |  | Tiempo  (segundos) |
| Completo | Convexo | Euleriano | Par | Impar |
| 1 | 1 | 5 | No | Si | Si | 3 | 2 | 0.003 |
| 2 | 3 | 10 | No | Si | No | 4 | 6 | 0.005 |
| 3 | 5 | 20 | No | Si | No | 8 | 12 | 0.028 |
| 4 | 10 | 30 | No | Si | No | 16 | 14 | 0.05 |
| 5 | 15 | 40 | No | Si | No | 18 | 22 | 0.087 |
| 6 | 20 | 50 | No | No | No | 26 | 24 | 0.125 |
| 7 | 25 | 60 | No | Si | No | 30 | 30 | 0.18 |
| 8 | 30 | 70 | No | No | No | 30 | 40 | 0.241 |
| 9 | 35 | 80 | No | Si | No | 40 | 40 | 0.34 |
| 10 | 40 | 90 | No | Si | No | 42 | 48 | 0.4 |
| 11 | 45 | 100 | No | Si | No | 50 | 50 | 0.479 |
| 12 | 50 | 150 | No | Si | No | 74 | 76 | 0.992 |
| 13 | 55 | 200 | No | Si | No | 100 | 100 | 1.728 |
| 14 | 60 | 250 | No | Si | No | 120 | 130 | 2.625 |
| 15 | 65 | 300 | No | Si | No | 142 | 158 | 3.746 |
| 16 | 70 | 350 | No | Si | No | 168 | 182 | 5.013 |
| 17 | 75 | 400 | No | Si | No | 194 | 206 | 6.61 |
| 18 | 80 | 450 | No | Si | No | 236 | 214 | 8.334 |
| 19 | 85 | 500 | No | Si | No | 250 | 250 | 10.114 |
| 20 | 90 | 550 | No | Si | No | 296 | 254 | 12.309 |

Conclusiones:

Mediante la realización de los múltiples experimentos registrados se pudo observar que es difícil que se genere un grafo completo. Se probó variar la semilla para generar los números aleatorios y obtener un grafo completo, pero al parecer sigue siendo difícil encontrar una combinación que nos permita obtener un grafo completo.

Independientemente de si los grafos generales eran completos o no se verifico si podían ser conexos o eulerianos, los resultados demuestran que es común que un grafo sea conexo si este mismo no es euleriano, aunque pueden darse casos donde el grafo, aunque sea conexo, pueda ser euleriano.

La mayor complicación durante este ejercicio fue generar las rutas de cada nodo de la mejor manera para que exista una gran variedad de rutas. Además de eso, el procedimiento más pesado fue calcular todas las rutas posibles de un nodo a otro para verificar si este puede ser convexo o no, a diferencia de los otros dos procedimientos que solo requieren leer la matriz de datos e identificar cuantas rutas impares existen.

Bibliografía:

<http://verso.mat.uam.es/~pablo.angulo/doc/laboratorio/b4s2.html>

[www.unipamplona.edu.co/unipamplona/portalIG/home\_23/recursos/.../grafo3.pdf](http://www.unipamplona.edu.co/unipamplona/portalIG/home_23/recursos/.../grafo3.pdf)

[www.ma.uva.es/~antonio/Industriales/Apuntes\_06-07/LabM/Grafos\_2007-2.pdf](http://www.ma.uva.es/~antonio/Industriales/Apuntes_06-07/LabM/Grafos_2007-2.pdf)

Ejercicio 2:

Introducción:

Transcriba el siguiente programa y agregue los siguientes módulos en funciones independientes: Calcule a) la media, b) la moda, c) la desviación estándar, d) Cuantos valores son pares e impares, e) cuales valores son primos, f) Cuales valores son amigos, g) la varianza y h) sume todos los elementos del arreglo.

**#include** <stdio.h>

**#include** <stdlib.h>

**#include** <math.h>

**#define** RANDMAX 10000

**int** **main**(**void**) {

**int** num=100, i=0, j=0, aux;

**long** **int** numeros[num];

**puts**("!!!Hello Students!!!");

**puts**("METODO DE LA BURBUJA");

//puts("Cuantos Numeros quieres:");

//scanf("%d",&num);

// LLENAR ARREGLO CON VALORES ALEATORIOS

**for** (i=0; i < num; i++)

{

numeros[i]= **rand**();

**printf**("%d, ",numeros[i]);

}

//ORDENAR

**for** (i=0; i<num;i++)

**for** (j=i+1;j<num;j++)

{

**if** (numeros[j] < numeros[i])

{aux = numeros[i];

numeros[i] = numeros[j];

numeros[j]=aux;}

}

**printf** ("\n");

**for** (i=0; i < num; i++)

**printf**("%d, ",numeros[i]);

**printf** ("\n");

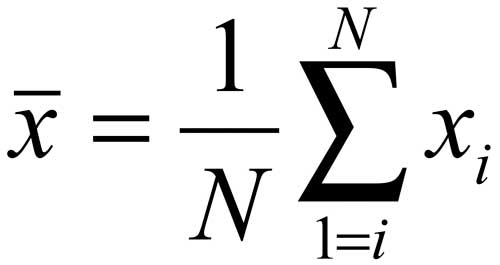
**printf** ("El elemento mas pequeño es: %d ", numeros[0]);

**printf** ("\n El elemento mas grande es: %d", numeros[num-1]);

**return** EXIT\_SUCCESS;

}

Marco teórico:

Media:

La media de un conjunto de números, algunas ocasiones simplemente llamada el promedio, es la suma de los datos dividida entre el número total de datos.

Moda:

La moda de un conjunto de datos es el dato que más veces se repite, es decir, aquel que tiene mayor frecuencia absoluta. Se denota por Mo. En caso de existir dos valores de la variable que tengan la mayor frecuencia absoluta, habría dos modas. Si no se repite ningún valor, no existe moda.

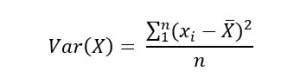
Desviación estándar:

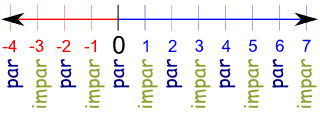
La desviación típica o desviación estándar (denotada con el símbolo σ o s, dependiendo de la procedencia del conjunto de datos) es una medida de dispersión para variables de razón (variables cuantitativas o cantidades racionales) y de intervalo. Se define como la raíz cuadrada de la varianza de la variable.

$ \sigma =\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i-\mu)^2$

La desviación estándar es la raíz cuadrada de la varianza. Es una de las medidas de dispersión, una medida que es indicativa de como los valores individuales pueden diferir de la media.

Varianza:

La unidad de medida de la varianza será siempre la unidad de medida correspondiente a los datos, pero elevada al cuadrado. La varianza siempre es mayor o igual que cero. Al elevarse los residuos al cuadrado es matemáticamente imposible que la varianza salga negativa. Y de esa forma no puede ser menor que cero.

Números pares e impares:

En las matemáticas y especialmente en la aritmética, un número par es un número entero que es divisible entre dos. ​ Se trata de un número entero que se puede escribir de la forma: 2k (es decir, divisible de manera entera entre 2), donde k es un entero (los números pares son los múltiplos del número 2).

Los números enteros que no son pares se llaman números impares (o números menores), y pueden escribirse como 2k+1.2​

Números primos:

Un número primo es un número entero mayor que cero, que tiene exactamente dos divisores positivos. También podemos definirlo como aquel número entero positivo que no puede expresarse como producto de dos números enteros positivos más pequeños que él, o bien, como producto de dos enteros positivos de más de una forma. Conviene observar que con cualquiera de las dos definiciones el 1 queda excluido del conjunto de los números primos.

Ejemplos:

* El 7 es primo. Sus únicos divisores son 1 y 7. Sólo puede expresarse como producto de 7·1.
* El 15 no es primo. Sus divisores son 1, 3, 5 y 15. Puede expresarse como 3·5. (y también como 15·1)

Los 25 primeros números primos son 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 y 97, que son todos los primos menores que 100.

Test de primalidad:

La cuestión de la determinación de si un número n dado es primo es conocida como el problema de la primalidad. Un test de primalidad (o chequeo de primalidad) es un algoritmo que, dado un número de entrada n, no consigue verificar la hipótesis de un teorema cuya conclusión es que n es compuesto.

Pseudocodigo:

**function** is\_prime(n)

**if** n ≤ 1

**return** false

**else if** n ≤ 3

**return** true

**else if** n mod 2 = 0 **or** n mod 3 = 0

**return** false

**let** i ← 5

**while** i \* i ≤ n

**if** n mod i = 0 **or** n mod (i + 2) = 0

**return** false

i ← i + 6

**return** true

Números amigos:

Dos números amigos son dos números enteros positivos a y b tales que la suma de los divisores propios de uno es igual al otro número y viceversa, es decir σ(a)=b y σ(b)=a, donde σ(n) es igual a la suma de los divisores de n, sin incluir a n. (La unidad se considera divisor propio, pero no lo es el mismo número.).

Un ejemplo es el par de naturales (220, 284), ya que:

* los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, que suman 284;
* los divisores propios de 284 son 1, 2, 4, 71 y 142, que suman 220.

Si un número es amigo de sí mismo (es igual a la suma de sus divisores propios), recibe entonces el nombre de número perfecto.

Primeros números amigos:

(220,284), (1184,1210),(2620,2924)

Metodología:

Conjunto de datos: 1  
**Datos:** 2, 8, 5, 1, 10, 5, 9, 9, 3, 5, 6, 6, 2, 8, 2, 2, 6, 3, 8, 7,

**Datos ordenados:** 1, 2, 2, 2, 2, 3, 3, 5, 5, 5, 6, 6, 6, 7, 8, 8, 8, 9, 9, 10,

**a) Media**:

Formula:

Media = (1 + 2 + 2 + 2 + 2 + 3 + 3 + 5 + 5 + 5 + 6 + 6 + 6 + 7 + 8 + 8 + 8 + 9 + 9 + 10)/20 = 107/20 = 5.35

**b) Moda**:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Total |
| Repeticiones | 1 | 4 | 2 | 0 | 3 | 3 | 1 | 3 | 2 | 1 | 20 |

Valor que más se repite: 2

Moda = 2

**c) Desviación estándar**:

Formula:

Desviación estándar =

Sqrt(((1-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 +

(3-5.35)^2 + (3-5.35)^2 + (5-5.35)^2 + (5-5.35)^2 + (5-5.35)^2 +

(6-5.35)^2 + (6-5.35)^2 + (6-5.35)^2 + (7-5.35)^2 + (8-5.35)^2 +

(8-5.35)^2 + (8-5.35)^2 + (9-5.35)^2 + (9-5.35)^2 + (10-5.35)^2) / 20) =

Sqrt((148.550003)/20) = 2.7253

**d) ¿Cuántos valores son pares e impares?**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Datos** | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Par/Impar** | impar | par | impar | impar | par | impar | par | impar | par |
| **Repeticiones** | 1 | 4 | 2 | 3 | 3 | 1 | 3 | 2 | 1 |

Pares: 4 + 3 + 3 + 1 = 11

Impares= 1 + 2 + 3 + 1 + 2 = 9

Total = 11 + 9 = 20

**e) ¿Cuáles valores son primos?**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Datos** | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Primo** | No | Si | Si | Si | No | Si | No | No | No |
| **Repeticiones** | 1 | 4 | 2 | 3 | 3 | 1 | 3 | 2 | 1 |

**Test de primalidad:**

**n = 1**

function is\_prime(n)

**if n ≤ 1**

**return false**

else if n ≤ 3

return true

else if n mod 2 = 0 or n mod 3 = 0

return false

let i ← 5

while i \* i ≤ n

if n mod i = 0 or n mod (i + 2) = 0

return false

i ← i + 6

return true

**n = 5**

function is\_prime(n)

if n ≤ 1

return false

else if n ≤ 3

return true

else if n mod 2 = 0 or n mod 3 = 0

return false

let i ← 5

while i \* i ≤ n

if n mod i = 0 or n mod (i + 2) = 0

return false

i ← i + 6

**return true**

**n = 8**

function is\_prime(n)

if n ≤ 1

return false

else if n ≤ 3

return true

else if **n mod 2 = 0** or n mod 3 = 0

**return false**

let i ← 5

while i \* i ≤ n

if n mod i = 0 or n mod (i + 2) = 0

return false

i ← i + 6

return true

En cada caso de prueba se resaltó el valor, la condición que cumplió y el resultado obtenido.

**f) ¿Cuáles valores son amigos?**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Datos** | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Amigos** | No | No | No | No | No | No | No | No | No |
| **Repeticiones** | 1 | 4 | 2 | 3 | 3 | 1 | 3 | 2 | 1 |

Datos: 9 y 10

Divisores de 9:

9%1=0 Divisor

9%2=1

9%3=0 Divisor

9%4=1

9%8=1

Divisores de 10:

10%1=0 Divisor

10%2=0 Divisor

10%4=0 Divisor

Suma de divisores 9: 1 + 3 = 4

Suma de divisores 10: 1 + 2 + 5 = 8

4 es distinto que 10

8 es distinto que 9

No son amigos

**g) Varianza**:

Formula:

Desviación estándar =

((1-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 + (2-5.35)^2 +

(3-5.35)^2 + (3-5.35)^2 + (5-5.35)^2 + (5-5.35)^2 + (5-5.35)^2 +

(6-5.35)^2 + (6-5.35)^2 + (6-5.35)^2 + (7-5.35)^2 + (8-5.35)^2 +

(8-5.35)^2 + (8-5.35)^2 + (9-5.35)^2 + (9-5.35)^2 + (10-5.35)^2) / 20 =

(148.550003/20) = 7.4275

**h) Sume todos los elementos del arreglo**:

Formula:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Repeticiones | 1 | 4 | 2 | 0 | 3 | 3 | 1 | 3 | 2 | 1 |

Suma = 1 + 2 + 2 + 2 + 2 + 3 + 3 + 5 + 5 + 5 + 6 + 6 + 6 + 7 + 8 + 8 + 8 + 9 + 9 + 10 = 107

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 | Datos | Menor | Mayor | Suma | Media | Moda | Varianza | Desviación estándar | Par | Impar | Primos | Amigos | Tiempo (segundos) |
| 1 | 1 | 5 | 41 | 26500 | 70511 | 14102.2 | 41 | 91298328 | 9555.015 | 2 | 3 | 41 |  | 0.04 |
| 2 | 3 | 10 | 48 | 27217 | 136827 | 13682.7 | 48 | 70853368 | 8417.44 | 5 | 5 | 9091 |  | 0.19 |
| 3 | 5 | 100 | 54 | 32640 | 1678103 | 16781.02 | 54 | 9525.77 | 9074080 | 47 | 53 | 1231,2633,4801,7253,12517,20107,24793,28211,31193 |  | 0.992 |
| 4 | 10 | 200 | 64 | 32719 | 3307522 | 16537.6 | 64 | 9288.46 | 86275544 | 94 | 106 | 71,313,647,1427,2441,2557,4363,7963,9413,11351,11923,13697,15349,16607,18133,18593,20929,21563,22111,25601,28541,29717,32159,32327,32719 |  | 3.824 |
| 5 | 15 | 300 | 53 | 32567 | 5032351 | 16774.5 | 10387 | 9588.85 | 91946224 | 149 | 151 | 53,167,283,1367,2441,2473,6203,8317,8467,9227,9491,9787,10079,10771,11621,12491,12893,13177,14081,16183,16673,17093,17569,18217,20551,20849,22063,23917,24151,25679,25693,26849,27283,28547,29311,29437,30313,31247,32257 |  | 8.735 |
| 6 | 20 | 500 | 42 | 32610 | 8025276 | 16050.5 | 6002 | 9638.23 | 92895568 | 252 | 248 | 103,1049,1213,1471,1699,2111,2131,3527,3803,4217,6547,6547,7213,8161,8363,8681,9973,10177,12109,15073,15401,15803,16871,17317,17957,17989,18043,18553,18637,20233,20357,20483,21487,22921,24683,24749,25057,25693,26729,28393,28463,28477,28867,32381 |  | 23.192 |

Debido a la cantidad de números generados no es posible mostrar todos los números primos resultantes.

Conclusiones:

Durante este ejercicio se buscó realizar funciones independientes que utilizan el mismo conjunto de datos para realizar distintos cálculos. Esto en un futuro debería permitir paralelizar el ejercicio en función a las operaciones.

Sin embargo, se observó que algunos cálculos pueden requerir de cálculos previos como calcular el promedio que requiere primero calcular la suma total o calcular la desviación estándar que requiere del cálculo de la varianza. En cada método se pueden repetir los cálculos, sin embargo, esto podría alentar el rendimiento del programa.

Un cálculo que es difícil de demostrar en la ejecución del programa es el cálculo de números amigos, esto es porque solo una cierta cantidad de números cumplen con esa condición y la generación de números aleatorios no permite obtener con frecuencia estos números necesarios. Es por esto que en la experimentación no se muestra ningún resultado sobre números amigos.

La única manera de obtener resultados sobre este cálculo es modificar manualmente el conjunto de datos para que contenga almenas dos números amigos y poder comprobar que el programa está correcto.

Bibliografía:

<https://es.wikipedia.org/wiki/Desviaci%C3%B3n_t%C3%ADpica>

<http://www.alcula.com/es/calculadoras/estadistica/desviacion-estandar/>

<http://economipedia.com/definiciones/varianza.html>

<https://www.varsitytutors.com/hotmath/hotmath_help/spanish/topics/mean-median-mode>

<https://www.portaleducativo.net/octavo-basico/790/Media-moda-mediana-rango>

<http://mimosa.pntic.mec.es/jgomez53/matema/conocer/primos.htm>

[www.vaxasoftware.com/doc\_edu/mat/numamigos\_esp.pdf](http://www.vaxasoftware.com/doc_edu/mat/numamigos_esp.pdf)

<https://es.wikipedia.org/wiki/N%C3%BAmeros_amigos>

<https://en.wikipedia.org/wiki/Primality_test>

<http://diagramas-de-flujo.blogspot.com/2013/01/determinar-si-dos-numeros-son-amigos-en-Cpp.html>