



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

INGENIERIA EN SISTEMA COMPUTACIONALES

UNIDAD DE APRENDIZAJE: ANALISIS Y DISEÑO DE ALGORITMOS

PRÁCTICA 1:

ALGORITMOS ITERATIVOS

ALUMNOS:

BASTIDA GONZÁLEZ MARÍA GUADALUPE

2021630164

MEJÍA ROMO PABLO EMANUEL

2020350970

PROFESORA: LUZ MARIA SANCHEZ GARCIA

PRÁCTICA 1

ALGORITMOS DE ORDENAMIENTO ITERATIVOS

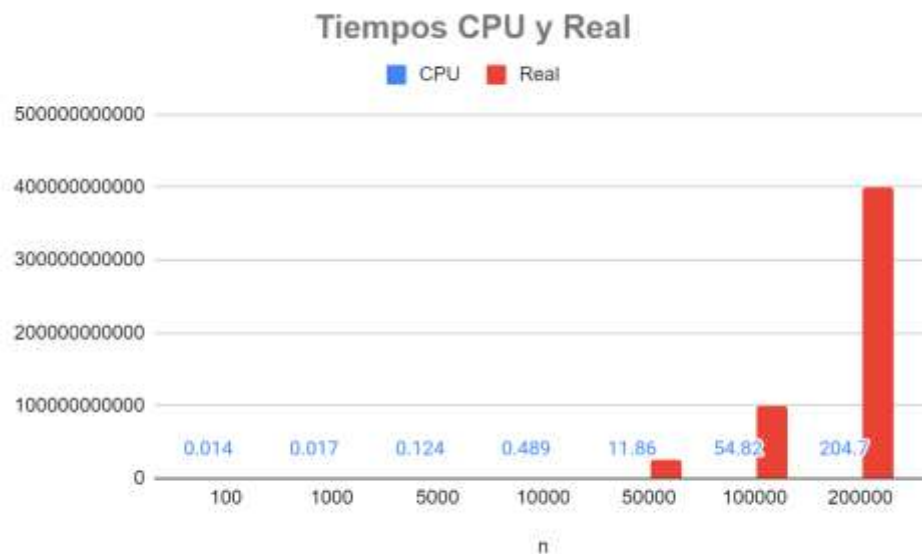
Planteamiento de problema:

Análisis de los algoritmos de ordenamiento iterativo: Burbuja, Burbuja bidireccional, Inserción, Selección y Shell a través de su complejidad temporal y comprobación experimental.

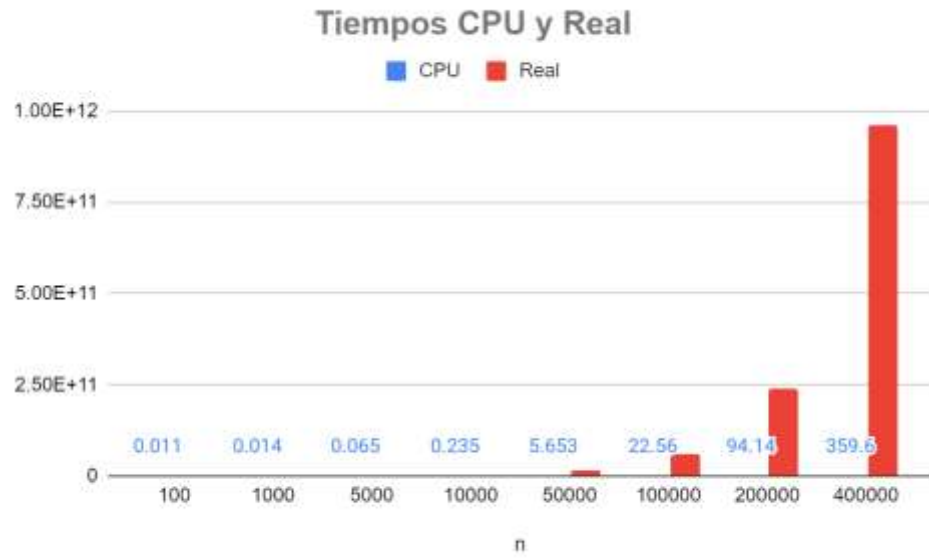
Actividades.

1. Programar en ANSI C, cada uno de los algoritmos de ordenamiento mencionados.
2. El programa deberá ser capaz de recibir un parámetro que indica el número de enteros a ordenar a partir de un archivo con máximo 10,000,000 números en desorden.
3. Medir el tiempo que tarda cada algoritmo en ordenar el archivo completo (10,000,000) y compare los tiempos (real y de CPU) de cada algoritmo gráficamente con una gráfica de barras.

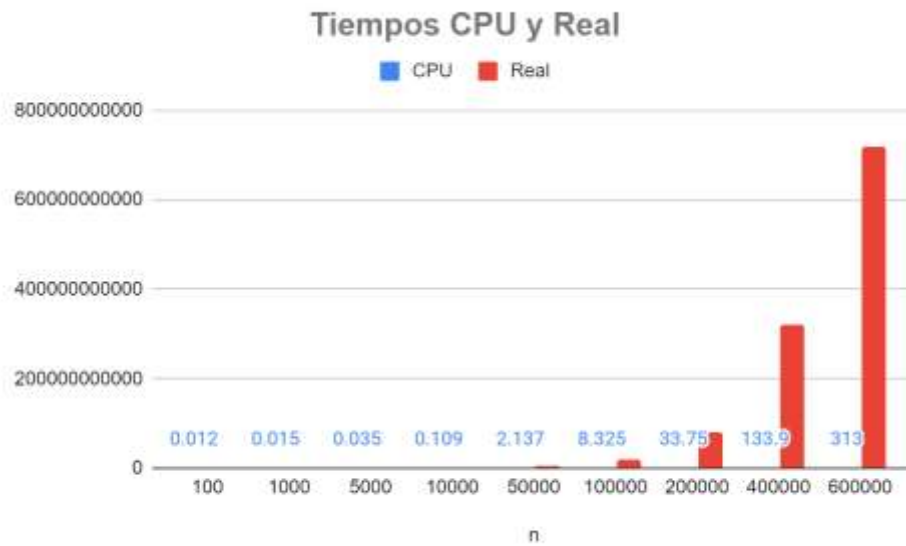
- **Burbuja:**



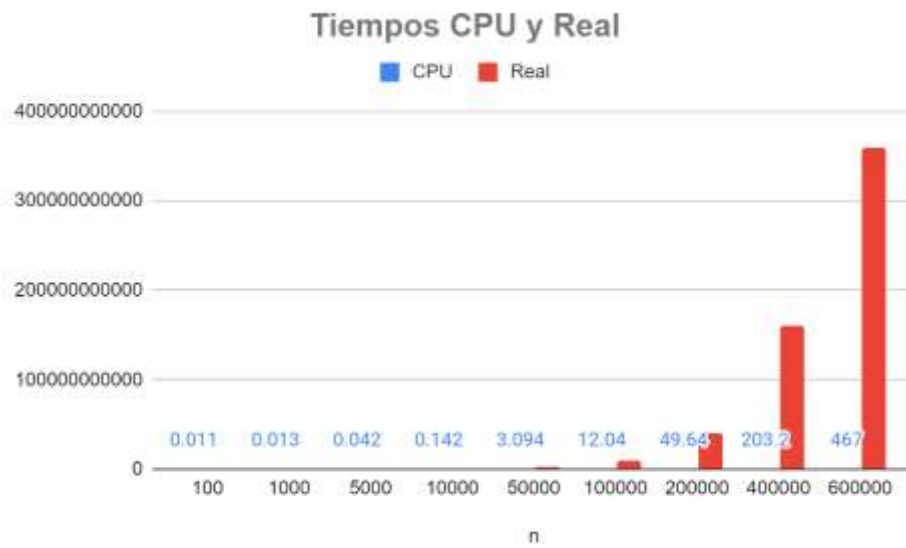
- **Burbuja bidireccional:**



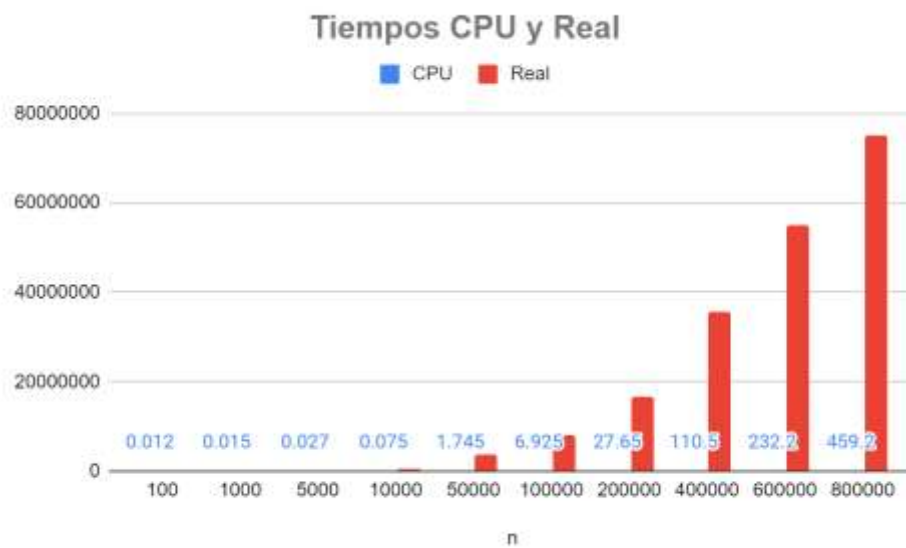
- **Inserción:**



- Selección:



- Shell:



4. Realizar un análisis temporal para cada algoritmo, ordenando. Los primeros 100 1,000 5,000 10,000 50,000 100,000 200,000 400,000 600,000 800,000 1,000,000 2,000,000 3,000,000 4,000,000 5,000,000 6,000,000 7,000,000 8,000,000 9,000,000 y 10,000,000.

- **Burbuja:**

$$T(n) = 10n^2 + 10n + 6$$

- **Burbuja bidireccional:**

$$T(n) = 6n^2 + 10n + 3$$

- **Inserción:**

$$T(n) = 2n^2 + 6n + 2$$

- **Selección:**

$$T(n) = 20n^2 + 26n + 11$$

- **Shell:**

$$T(n) = 8n \log \frac{n^2}{4} + 8 \log \frac{n^2}{4} + 2 \log \frac{n}{2} + \log n + 4n + 4$$

5. Realizar una aproximación polinomial del comportamiento temporal de cada uno de los algoritmos probados según el punto 4 en una tabla:

Burbuja

K	Instrucciones
100	101006.00
1000	10010006.00
5000	250050006.00
10000	1000100006.00
50000	25000500006.00
100000	100001000006.00
200000	400002000006.00
400000	1600004000006.00
600000	3600006000006.00
800000	6400008000006.00
1000000	10000010000006.00
2000000	40000020000006.00
3000000	90000030000006.00
4000000	160000040000006.00
5000000	250000050000006.00
6000000	360000060000006.00
7000000	490000070000006.00
8000000	640000080000006.00
9000000	810000090000006.00
10000000	1000000100000010.00

Burbuja bidireccional

k	Instrucciones
100.00	61003.00
1000.00	6010003.00
5000.00	150050003.00
10000.00	600100003.00
50000.00	15000500003.00
100000.00	60001000003.00
200000.00	240002000003.00
400000.00	960004000003.00
600000.00	2160006000003.00
800000.00	3840008000003.00
1000000.00	6000010000003.00
2000000.00	24000020000003.00
3000000.00	54000030000003.00
4000000.00	96000040000003.00
5000000.00	150000050000003.00
6000000.00	216000060000003.00
7000000.00	294000070000003.00
8000000.00	384000080000003.00
9000000.00	486000090000003.00
10000000.00	600000100000003.00

Inserción

K	Instrucciones
100.00	20602.00
1000.00	2006002.00
5000.00	50030002.00
10000.00	200060002.00
50000.00	5000300002.00
100000.00	20000600002.00
200000.00	80001200002.00
400000.00	320002400002.00
600000.00	720003600002.00
800000.00	1280004800002.00
1000000.00	2000006000002.00
2000000.00	8000012000002.00
3000000.00	18000018000002.00
4000000.00	32000024000002.00
5000000.00	50000030000002.00
6000000.00	72000036000002.00
7000000.00	98000042000002.00
8000000.00	128000048000002.00
9000000.00	162000054000002.00
10000000.00	200000060000002.00

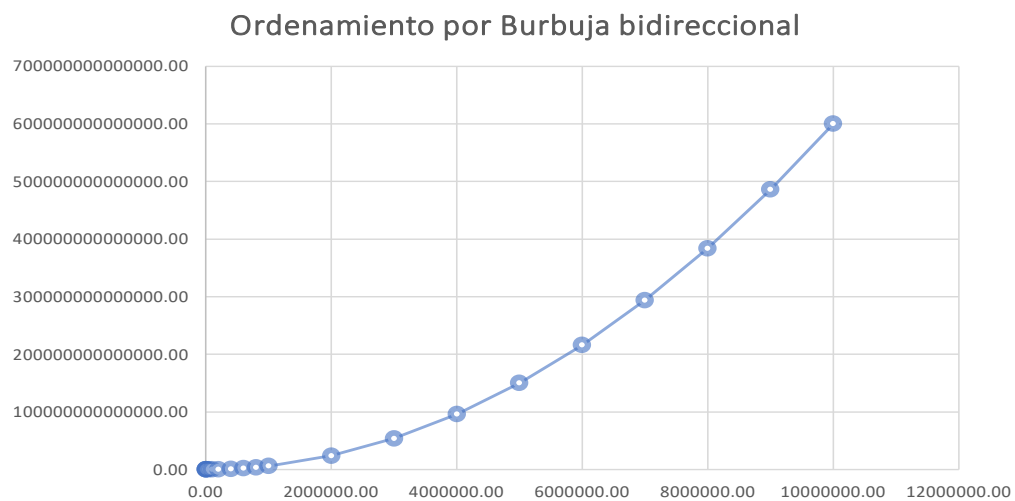
Selección

K	Instrucciones
100.00	202611.00
1000.00	20026011.00
5000.00	500130011.00
10000.00	2000260011.00
50000.00	50001300011.00
100000.00	200002600011.00
200000.00	800005200011.00
400000.00	3200010400011.00
600000.00	7200015600011.00
800000.00	12800020800011.00
1000000.00	20000026000011.00
2000000.00	80000052000011.00
3000000.00	180000078000011.00
4000000.00	320000104000011.00
5000000.00	500000130000011.00
6000000.00	720000156000011.00
7000000.00	980000182000011.00
8000000.00	1280000208000010.00
9000000.00	1620000234000010.00
10000000.00	2000000260000010.00

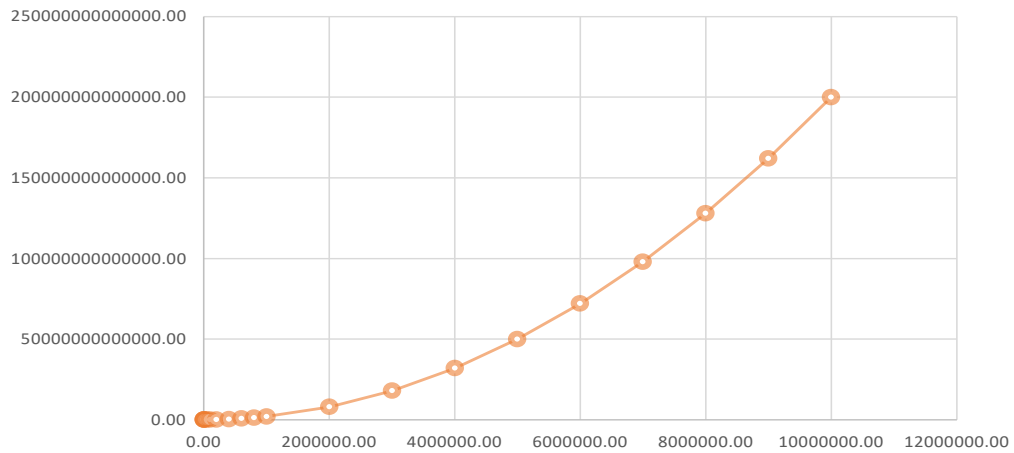
Shell

K	Instrucciones
100.00	9542.40
1000.00	147627.90
5000.00	923236.46
10000.00	2006272.45
50000.00	11887994.96
100000.00	25375726.34
200000.00	53951170.10
400000.00	114302038.62
600000.00	177068539.26
800000.00	241403756.66
1000000.00	306905461.81
2000000.00	645810577.92
3000000.00	996793886.18
4000000.00	1355620791.14
5000000.00	1720280141.95
6000000.00	2089587396.54
7000000.00	2462759849.07
8000000.00	2839241198.57
9000000.00	3218615499.12
10000000.00	3600559894.09

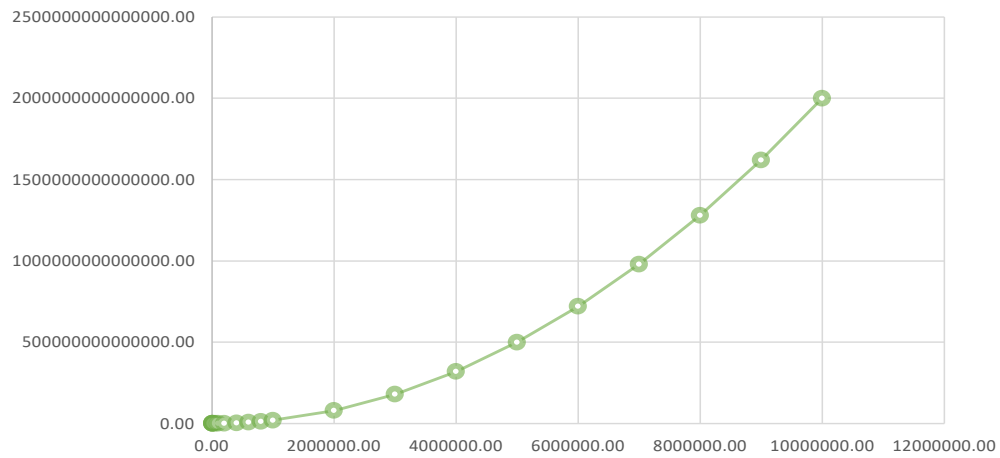
6. Graficar el comportamiento temporal de cada algoritmo (1 gráfica para cada uno de los algoritmos).



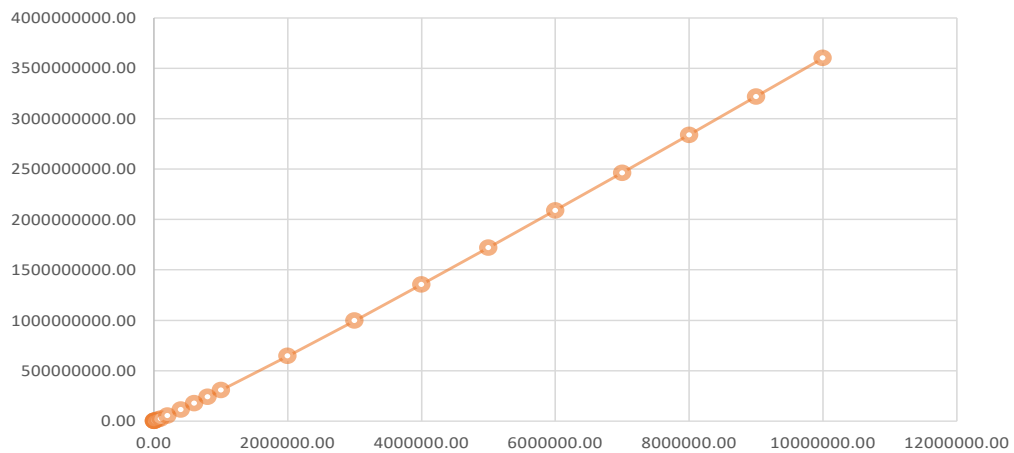
Ordenamiento por Inserción



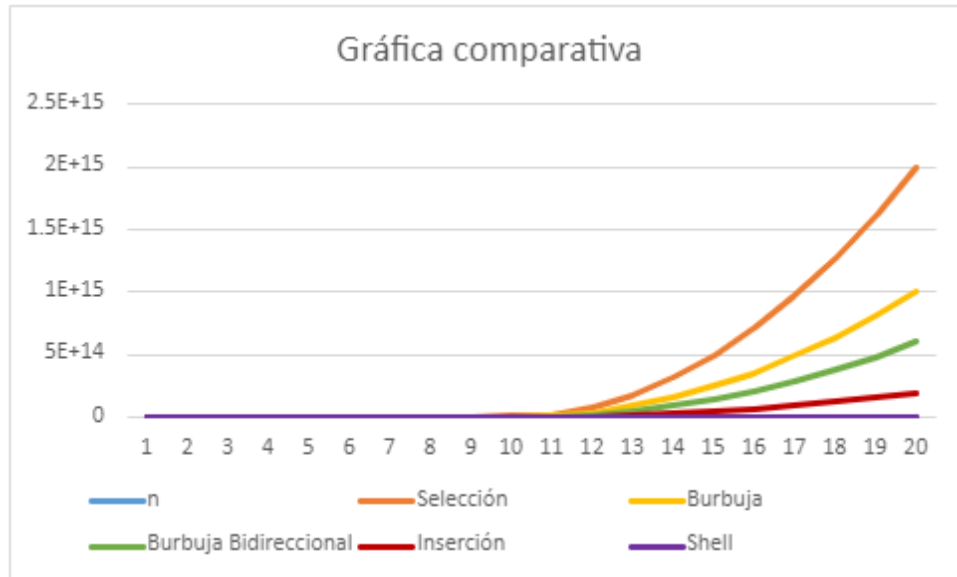
Ordenamiento por selección



Ordenamiento por Shell



7. Mostrar gráficamente la comparativa de las aproximaciones para cada algoritmo en una sola gráfica y determinar de manera justificada cuál es el más eficiente.



Como podemos observar en la gráfica, se muestra los diferentes comportamientos de los algoritmos, utilizando las complejidades temporales calculadas anteriormente, tenemos los siguientes comportamientos:

Algoritmo de ordenamiento	Orden
Burbuja	Cuadrático n^2
Burbuja Bidireccional	Cuadrático n^2
Inserción	Cuadrático n^2
Selección	Cuadrático n^2
Shell	$n \log n$

Por lo tanto tomando en cuenta, el número de comparaciones que realiza cada algoritmo conforme aumenta n , tenemos que el más eficiente es el algoritmo de ordenamiento Shell.

8. ¿Cuáles fueron las operaciones básicas que se tomaron para el cálculo de la complejidad temporal?

Para calcular la complejidad temporal se tomaron en cuenta, los, incrementos, comparaciones y asignaciones que se encontraban dentro de los ciclos como los for y while, así mismo se tomaron en cuenta las condiciones de los if.

9. Determine con base en las aproximaciones obtenidas cual será el tiempo real de cada algoritmo para 50,000,000 100,000,000 500,000,000 1,000,000,000 y 5,000,000,000 de números a ordenar.

n	Selección	Burbuja	Burbuja Bidireccional	Inserción	Shell
50000000	1.5E+16	2.5E+16	1.5E+16	5E+15	6118352152
100000000	6E+16	1E+17	6E+16	2E+16	12718352158
500000000	1.5E+18	2.5E+18	1.5E+18	5E+17	69183520233
1000000000	6E+18	1E+19	6E+18	2E+18	1.43184E+11
5000000000	1.5E+20	2.5E+20	1.5E+20	5E+19	7.71835E+11

10. Finalmente responda a las siguientes preguntas

A. ¿Cuál de los 5 algoritmos es más fácil de implementar?

Ordenamiento por inserción

B. ¿Cuál de los 5 algoritmos es el más difícil de implementar?

Ordenamiento por Shell

C. ¿Cuál algoritmo tiene menor complejidad temporal?

Ordenamiento por inserción

D. ¿Cuál algoritmo tiene mayor complejidad temporal?

Ordenamiento por Shell

E. ¿Cuál algoritmo tiene menor complejidad espacial? ¿Por qué?

Ordenamiento por inserción, porque únicamente utiliza 3 variables para efectuar el ordenamiento, además del arreglo que almacena los datos.

F. ¿Cuál algoritmo tiene mayor complejidad espacial? ¿Por qué?

Ordenamiento por burbuja bidireccional, porque ocupa 2 variables en las cuales se tiene el rango en el que se hará el recorrido; una más para determinar el sentido del recorrido; una que ocupa como valor booleano, y la variable que hace el acceso al arreglo.

G. ¿El comportamiento experimental de los algoritmos era el esperado? ¿Por qué?

Si, porque anteriormente se analizaron los algoritmos de forma teórica y al momento de llevarlos a la práctica, se pudo comprobar que ese análisis era correcto.

H. ¿Sus resultados experimentales difieren mucho de los del resto de los equipos? ¿A qué se debe?

Las respuestas, de los demás equipos no difieren mucho de los nuestros, puesto que casi tenemos los mismos resultados en las pruebas experimentales, las leves variaciones se deben en su mayoría a la diferencia de entornos técnicos, como lo es nuestro equipo de cómputo o hasta el mismo IDE que usamos.

I. ¿Existió un entorno controlado para realizar las pruebas experimentales? ¿Cuál fue?

Si, pues al momento de utilizar el IDE DEV c++, se ahorraron instrucciones dentro del código, porque a diferencia de otros, los ejecutables que compila este IDE muestran el tiempo de ejecución total del programa.

Además, los resultados obtenidos se tomaron de un solo equipo con las siguientes especificaciones:

- Procesador: Intel Core i7-4600 de 4 núcleos a 2.69 GHz
- Memoria RAM: DDR3 de 8 GB
- Unidad de almacenamiento: Unidad de estado sólido de 500 GB
- Sistema operativo: x64 o 64 bits
- Compilador: MinGw32-gcc v6.3.0-1
- IDE: Dev C++ v5.11

Además, se procuró que el equipo no ejecutara otros programas mas que el IDE y el mismo algoritmo.

J. ¿Qué recomendaciones darían a nuevos equipos para realizar esta práctica?

Segun nuestra experiencia realizando esta práctica, algunas de las recomendaciones que deberian tomar antes de realizarla son:

- Tener presente el procedimiento para el cálculo de la complejidad temporal de un algoritmo.
- Saber programar en C.
- Conocer las especificaciones de tu equipo de cómputo.
- Investigar el estándar ANSI C para el lenguaje de programación C.