# **UNIVERSIDAD ICESI**

# *Facultad de Ingeniería*

*Proyecto Integrador*

*I Periodo de 2019*

# **Experimento de Ordenamiento de los Algoritmos Shellsort e Insertion Sort.**

**Miguel Angel Romero, Alejandra González Vélez, Angie Valentina Córdoba**

*Ingeniería de Sistemas, Facultad de ingeniería, Universidad Icesi, Cali Colombia*

Recibido: 11 de septiembre de 2019

Introducción:

En este experimento de campo lo que se busca es estudiar y evaluar los resultados de un proceso de análisis computacional con respecto a lo relacionado con el tiempo de ordenamiento de los algoritmos Shell Sort e Insertion Sort. Por otra parte, se busca identificar los elementos relevantes que soportan la elección e implementación de la mejor solución en particular. El objetivo primordial de este análisis es estudiar el tiempo de ejecución como función del “tamaño” de los datos de entrada. Para ello se usarán varias técnicas de evaluación, tales como; Medición tiempos de ejecución de los programas con datos de entrada de distintos tamaños, dependiendo de otros factores que se mencionan en el transcurso de este informe y contar el número de operaciones que el programa está realizando.

Planeación y Realización:

La complejidad que pueda tener un algoritmo de ordenamiento representa la cantidad de recursos de tiempo que son necesarios para el algoritmo resolver el problema de ordenamiento. En consecuencia, este factor será el que va a determinar la eficiencia de dicho algoritmo. No obstante, cabe resaltar que, el mismo algoritmo en diferentes máquinas de cómputo tarda más que en otras máquinas, y esto se debe más que todo a factores de ejecución de tareas que las otras computadoras están llevando a cabo, si la memoria RAM es lo suficiente cuando se vaya a ejecutar el algoritmo de ordenamiento, entre otras.

El tiempo se calcula por el coste del tamaño de los datos, tomando el esperado, el promedio y el mejor coste de tamaño. Cabe resaltar que hay muchos algoritmos que nos permiten ordenar una secuencia de datos, en este proyecto de investigación nos vamos a limitar únicamente a analizar y estudiar los algoritmos de ordenamiento SHELL SORT e INSERTION SORT.

La variable de respuesta para este experimento es el tiempo. Científicos en computación emplean una forma de categorizar y comparar los algoritmos, de tal suerte que se pueda categorizar de forma rápida el rendimiento de un algoritmo. Esta forma de medir el tiempo de ejecución de un algoritmo se denomina notación Gran O ó big-O para determinar de forma aproximada la complejidad de un algoritmo en función de la variable de entrada.

|  |  |
| --- | --- |
| Algoritmo | Notación Big-O |
| Shell Sort | O( |
| Insertion Sort | O( |

Para este caso, se utilizarán 3 factores estudiados como los más relevantes en el experimento, estos son:

1. Algoritmo de ordenamiento
2. Tamaño del arreglo
3. Procesador

Todos estos ejercen cierta influencia significativa sobre el problema, por lo cual es importante tenerlos en cuenta en el momento de la evaluación. El algoritmo que se usa es relevante ya que cada uno tiene un tiempo de ejecución diferente, así como también es importante el tamaño del arreglo, pues dependiendo de qué tan grande sea el arreglo así mismo será el tiempo que se demore el algoritmo en ejecutarse. Finalmente, también se tiene en cuenta el procesador, ya que la capacidad de procesamiento de este podría influir en la velocidad del algoritmo.

Algunos de los factores no controlables identificados son: el estado de ordenamiento del arreglo y la cantidad de procesos en simultánea que se están ejecutando al momento que se está ejecutando el algoritmo.

Para cada factor estudiado se han determinado unos niveles, los cuales serán necesarios para realizar el experimento y medir los resultados que se obtengan. Los niveles determinados son los siguientes:

* Algoritmo de ordenamiento:
  + ShellSort
  + InsertionSort
* Tamaño del arreglo
* Procesador
  + intel core i5
  + intel core i7

La unidad experimental para cada tratamiento serán veinte(20) repeticiones, cada uno en las mismas condiciones y los resultados serán almacenados en una hoja electrónica.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabla de Tratamientos** | | | |
| **# De Tratamiento** | **Algoritmo de ordenamiento** | **Tamaño del arreglo** | **Procesador** |
| 1 | ShellSort |  | intel core i5 |
| 2 | ShellSort |  | intel core i7 |
| 3 | ShellSort |  | intel core i5 |
| 4 | ShellSort |  | intel core i7 |
| 5 | InsertionSort |  | intel core i5 |
| 6 | InsertionSort |  | intel core i7 |
| 7 | InsertionSort |  | intel core i5 |
| 8 | InsertionSort |  | intel core i7 |

La práctica de investigación “Tiempo de Ordenamiento de los Algoritmos Shell e Insertion” es realizada por los estudiantes de Ingeniería de Sistemas:

* Alejandra González
* Angie Valentina Córdoba
* Miguel Angel Romero

Esta práctica de investigación se dividió en varias partes; primero se procedió a elegir y escoger las características de calidad a ser evaluadas, dependiendo de los factores que se debían de tener en cuenta a la hora de hacer esta práctica de la manera más acertada posible; tales como:

* Algoritmos de Ordenamiento
* Tamaño del arreglo  
  Procesador

Después se procedió a escoger los niveles de referencia que se iban a tener para poder hacer la metodología de experimentación, la cual más delante se explicará de manera más detallada. Los niveles escogidos fueron los siguientes:

* Shellsort, InsertionSort
* ,
* i5 , i7

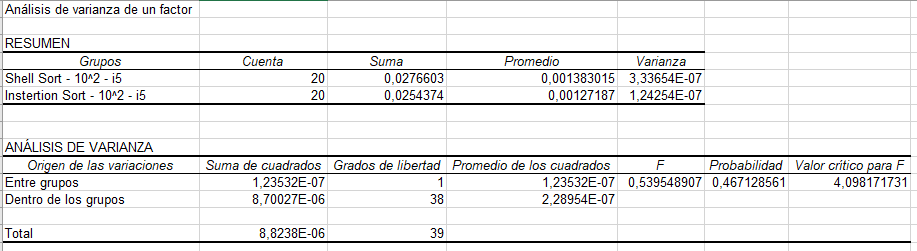
Estos niveles fueron escogidos teniendo en cuenta los factores estudiados. Una vez tanto los factores estudiados, como los niveles hayan sido escogido se procede a calcular los tratamientos previamente definidos. El número de tratamientos son 8 en total. Como cada uno de los integrantes de este trabajo de campo tiene un procesador diferente, se dividirán las tareas y el participante que tenga el procesador i5 hará los experimentos con la máquina correspondiente, el que tenga i7 con la máquina de cómputo correspondiente y así sucesivamente, para poder tener un resultado variado a la hora de analizar los datos. Los datos serán analizados haciendo uso de la herramienta de software EXCEL, la cual mostrará por medio de gráficas el comportamiento de cada uno de los algoritmos con su respectivo tratamiento en un punto de tiempo determinado.

Análisis e interpretación

Se realizó una prueba ANOVA, teniendo en cuenta la información recolectada por cada una de las muestras. Con cada una de las pruebas se busca determinar cuál es el mejor algoritmo, por lo que se han comparado los tiempos de ejecución para cada uno en las mismas condiciones.

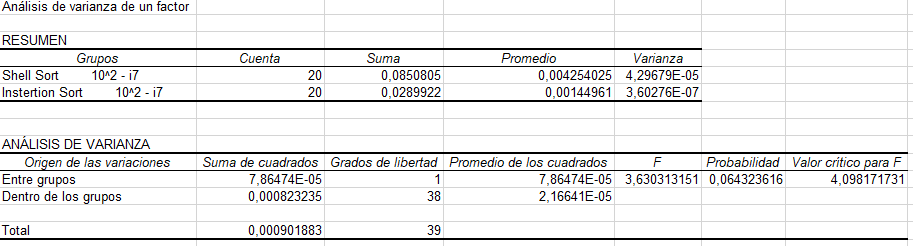
Los resultados fueron los siguientes:

Tratamiento 1 vs Tratamiento 5:



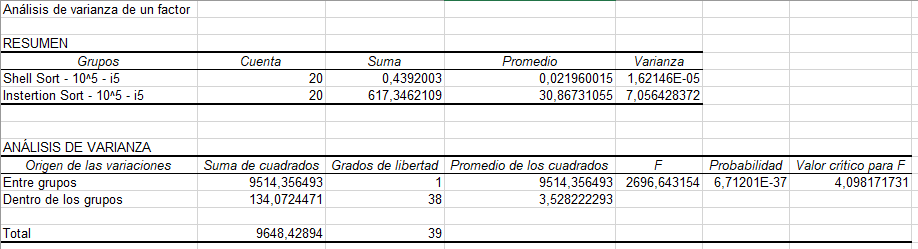
Como se puede notar en el análisis, el F crítico es mayor que el F calculado, por lo cual se puede concluir que no existe una diferencia significativa al ordenar un arreglo de posiciones, con un procesador core i5 utilizando el Shell sort ó Insertion sort.

Tratamiento 2 vs Tratamiento 6:



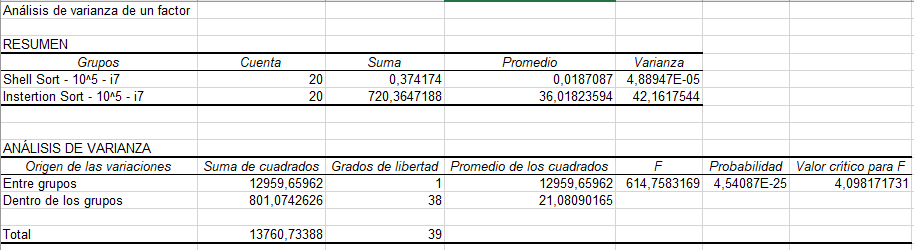
Como se puede notar en el análisis, el F crítico es mayor que el F calculado, por lo cual se puede concluir que no existe ninguna diferencia significativa al ordenar un arreglo de posiciones, con un procesador core i7 utilizando los algoritmos Shellsort o Insertion sort.

Tratamiento 3 vs tratamiento 7:



Como se puede notar en el análisis, el F crítico es menor que el F calculado, por lo que uno de los algoritmos tiene una media diferente. En este caso, se puede concluir que el mejor algoritmo es el Shell sort, pues el promedio del tiempo es menor al promedio de tiempo del algoritmo Insertion sort, ambos ejecutados con un procesador i5 y en arreglos de posiciones.

Tratamiento 4 vs Tratamiento 8:



Como se puede notar en el análisis, el F crítico es menor que el F calculado, por lo que uno de los algoritmos tiene una media diferente. En este caso, se puede concluir que el mejor algoritmo es el Shell sort, pues el promedio del tiempo es menor al promedio de tiempo del algoritmo Insertion sort, ambos ejecutados con un procesador i7 y en arreglos de posiciones.

Conclusión

En conclusión, al realizar el experimento y analizar los resultados, se puede notar que el mejor algoritmo depende en mayor medida del tamaño del arreglo, pues para arreglos pequeños, cualquiera de los dos algoritmos produce buenos resultados, mientras que con arreglos de gran tamaño el algoritmo Shell sort produce un mejor tiempo que el Insertion sort, siendo el primero el mejor algoritmo de ordenamiento para cualquier ocasión.

Referencias

* Fernando Gómez (2019). La notación Big-O. Recuperado de: <https://sites.google.com/site/fernandoagomezf/programacion-en-c/tips-de-programador-c/algoritmos/la-notacion-big-o>