

## Algoritmos de Ordenamiento y Búsqueda:

Tanto los algoritmos de ordenamiento como los de búsqueda son de suma importancia en una situación de problema de esta naturaleza ya que cuando se está lidiando con una gran cantidad de datos el ordenamiento manual es un trabajo muy extenso y casi interminable. Por lo que, con el uso y la selección de uno de los algoritmos más efectivos se podrá conseguir este resultado en cuestión de segundos. Por otra parte, el algoritmo de búsqueda es esencial al momento que el usuario necesita encontrar un dato particular o bien un rango de datos específico que le sea de utilidad. Por lo cual, la eficiencia siempre es una de las mayores prioridades que se debe dar en este tipo de situaciones donde no se sabe cuál es la magnitud de datos con la que se estará lidiando. En efecto, es también importante entender a qué se refiere el concepto de algoritmo el cual se entiende “como un procedimiento computacional definido que toma un valor o conjunto de valores de entrada para producir uno o diversos valores de salida”. En esta misma definición se puede apreciar que una característica esencial de los mismos es su computacionalidad y/o automatización lo cual le otorga efectividad al proceso. Es por esto, que también es necesario realizar un análisis del tiempo de ejecución de cada algoritmo para averiguar qué tan rápido es lo cual podrá garantizar un alto flujo del programa. Generalmente el factor que más influye en el tiempo de ejecución de un algoritmo corresponde al tamaño de la entrada del mismo el cuál generalmente se representa por  $N$ . A continuación, se muestra una tabla de los tiempos de ejecución aproximados para  $N=100$  datos según la complejidad del algoritmo:

$O(\log(N))$	$10^{-7}$ segundos
$O(N)$	$10^{-6}$ segundos
$O(N \cdot \log(N))$	$10^{-5}$ segundos
$O(N^2)$	$10^{-4}$ segundos
$O(N^3)$	3 minutos
$O(2^N)$	$10^{14}$ años.
$O(N!)$	$10^{142}$ años.

Los algoritmos de ordenamiento son el ejemplo más común ya que si bien se quiere llegar a un mismo orden de los elementos, existen distintos métodos para hacerlo donde claramente unos serán más efectivos que otros. Por ejemplo, si bien el

método más obvio de ordenar una lista sería identificar el mínimo elemento y colocarlo al inicio para subsecuentemente hallar el segundo más pequeño y colocarlo después y así sucesivamente... este algoritmo es proporcional a la cantidad de datos ya que deberá efectuar esa misma cantidad de comparaciones por lo que su complejidad es de  $O(N^2)$ . Por tal motivo, se puede considerar que no es un algoritmo tan efectivo cuando lo

comparamos con otros métodos de tipo Quicksort y Mergesort los cuales tienen la filosofía de “divide y vencerás” en donde se trata de hacer particiones en la lista hasta llegar a elementos únicos y luego ordenarlos. Por esa razón, dichos algoritmos tienen una complejidad de  $O(N \cdot \log(N))$ . Esto se debe a que la cantidad de operaciones que se requieren para ordenar con estos métodos es mucho menor ya que no depende de la cantidad de datos que se reciban de entrada, el proceso seguirá un mismo número de operaciones. (Soto, 2012)

En mi experiencia general pude darme cuenta de que existían algoritmos más sencillos de programar y entender su lógica pero no necesariamente estos eran los más eficientes. Por lo que, tras realizar esta actividad realmente entendí la importancia por optar por algoritmos más complicados en términos de recursividad en lugar de utilizar aquellos que me eran más sencillos de programar. Adicionalmente comprendí la importancia de saber programarlos ya que a pesar de que la librería <vector> contiene su propia función para ordenar sus elementos, el hecho de construirlos “manualmente” y estar consciente de que existen diferentes caminos para ordenarlos nos entrena para aprender algoritmos con funcionalidades más complejas y nos recuerda que siempre existe una manera más efectiva de realizar cualquier proceso. Esto se aplica tanto a la programación como a nuestra vida cotidiana y los algoritmos que construimos a diario. No obstante, esto me ha enseñado que si nos tomáramos el tiempo para pensar en maneras innovadoras de obtener los mismos resultados de manera más rápida, podríamos aprovechar más de nuestro día sin tener que invertirle tanto tiempo a hacer nuestros quehaceres diarios, sino hacerlos de una manera más inteligente. En otro aspecto, cabe recalcar que como programadores debemos saber que el 50% del tiempo, las computadoras se encargan de ordenar y organizar datos e información, por lo que es importante conocer estos procesos como parte de nuestra formación general como ingenieros en la avenida de computación.

### **Referencias:**

Soto, R. (31 de Diciembre de 2012). Recuperado el 12 de Septiembre de 2020, de <http://bitacora.ingenet.com.mx/wp-content/uploads/2012/12/girasol5.jpg>