



Instituto Tecnológico y de Estudios Superiores de Monterrey

Análisis de ecuaciones diferenciales (MA1033.1)

**Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos
Fundamentales, investigación y reflexión**

Daniel Pérez Rojas

Salvador Alejandro Gaytán Ibáñez A01730311

12 de septiembre del 2020

Algoritmos de ordenamiento:

Un algoritmo de ordenamiento le permite al programador, como bien lo dice su nombre, ordenar datos e información de una manera eficiente y rápida en base a un criterio de ordenamiento.

Existen distintos tipos de algoritmos de ordenamiento de los cuales se destacan los menos eficientes pero más sencillos de programar como lo son el ordenamiento por burbuja, por intercambio, por selección y por inserción, donde estos algoritmos más sencillos de ordenamiento cumplen la función de ordenar una cantidad N de datos, sin embargo no son muy eficientes ya que por su estructura tienen tiempos de ejecución muy elevados, es por esto que los algoritmos más complicados de comprender y programar son los más populares en la programación aplicada ya que su eficiencia y tiempos de ejecución son mucho menores a los previamente mencionados. Algunos ejemplos de métodos de ordenamiento más eficientes pero complicados son el Merge Sort y el Quicksort.

Algoritmos de búsqueda:

Por su parte, los algoritmos de búsqueda son exactamente lo que el nombre denota, algoritmos que permiten al programador buscar en una lista de información o datos ordenados un valor o rango de valores específicos y, al igual que los algoritmos de ordenamiento, existen distintos tipos de algoritmos de búsqueda donde algunos como el lineal (secuencial) si bien cumplen con el propósito de identificar un valor son poco eficientes comparados con algoritmos como la búsqueda binaria la cual es mucho más eficiente y cuenta con tiempos de ejecución extremadamente bajos a comparación con la búsqueda lineal.

Reflexión:

Para la elaboración de este proyecto (actividad 1.3) fue vital el uso de distintos métodos de ordenamiento y búsqueda ya que por la naturaleza de la misma era necesario manejar una inmensa cantidad de datos, ordenarlos y finalmente filtrarlos a discreción del usuario, es por esto que para poder ordenar los datos se optó por un método eficiente para reducir los tiempos de ejecución en gran medida el cual fue el Merge Sort, pero, en el caso de la búsqueda de datos se optó por un modelo lineal ya que por las características de la estructura del código realizado por mi equipo y por mi era el más adecuado, asimismo, los tiempos de ejecución en búsqueda contra los de ordenamiento son bastante pequeños por lo que utilizar un algoritmo un poco menos eficiente pero de una naturaleza más adecuada para el proyecto nos pareció una solución simple.

Bibliografías:

WH. (N/A). Algoritmos de ordenamiento. 12/09/2020, de IWH Sitio web:
http://lwh.free.fr/pages/algo/tri/tri_es.htm

Villalobos,J. (2006). Busqueda de vectores. 12/09/2020, de Unicauca Sitio web:
<http://artemisa.unicauca.edu.co/~nediaz/EDDI/cap02.htm>