Reflexión.

Actividad 1_3.

En un principio, considero que la actividad se veía bastante retadora, pues era la puesta en práctica de los últimos temas e implicaba, sobre todo, la buena toma de decisiones y un análisis de la situación para determinar cuales eran los algoritmos que mejor se ajustaban a nuestras necesidades y que las resolvieran de una forma lo suficientemente eficiente.

Para empezar, la creación de la estructura “Bitacora”. En un primer momento, nos pareció una mejor idea hacerla una clase, pero por el uso que se le iba a dar (no se requería de métodos ni herencias) hacer un struct era más práctico, conteniendo todos los datos de cada registro además de una variable extra deducida del mes y el día la cual sería llamada “key”, calculada de la siguiente manera:

$$key = Número_{mes} * 100 + dia$$

Esta última operación estaría a cargo de una función de nombre “createKey” pues más adelante volvería a ser utilizada.

Tras crear el vector de bitácora, llenarlo con el archivo y las key, hacía falta ordenarlo. Esta operación fue realizada mediante un mergeSort que recibía el vector principal, un vector auxiliar, el valor del inicio (0) y el valor fin (la cantidad total de registros menos 1); y reagrupaba los registros en base a las key de menor a mayor. La elección de este algoritmo de ordenamiento fue porque su peor caso era de orden (nlog n), lo cual lo convertía en el más eficiente de todos.

Para la búsqueda se utilizó la búsqueda binaria, de orden ($\log_2 n$). En un principio parecía mejor opción el usar la secuencial pues la búsqueda debía desplegar desde el primer valor correspondiente al inicio hasta el último de los finales, es decir, con todos los registros repetidos. Con vectores sin tantos datos podría no haber demasiado problema (aunque siga siendo poco eficiente), pero en este caso, debido al volumen de la búsqueda, no podíamos darnos el lujo de un orden (n^2), sería demasiado tardado y un trabajo mucho más pesado. Por ello mejor se adaptó la búsqueda en 2 funciones que buscaran en el inicio y el fin, de tal manera que no aumentaran el orden.

Me gustan mucho este tipo de actividades retadoras que te invitan a pensar más allá de simplemente aplicar los algoritmos aprendidos. Además, desenmarañar lo que hay detrás de las funciones de búsqueda y ordenamiento ya existentes abre un panorama más completo y me dio una mejor idea de lo importante que es trabajar respecto a la eficiencia de los programas