Reflexión

Para comenzar a programa esta tarea la primera tarea fue cargar los datos del archivo a un vector utilizando un ciclo while de complejidad O(n), donde n es la cantidad de líneas del archivo, luego utilizamos el vector creado para realizar un nuevo vector índice que contenga el valor total de la fecha, para facilitar el ordenamiento y búsqueda, comparando tan solo en el vector índice y realizando a la organización en ambos de los vectores. La complejidad en la función que realiza el índice es de O(n), teniendo dos ciclos constantes dentro de uno lineal donde se toman toma el mes y día y se crea un singular numero que le atribuye un valor único a cada fecha de la siguiente forma (mes*100) + día (ejemplo 7/28 =728). Inicialmente el algoritmo de ordenamiento utilizado fue uno de burbuja de complejidad O(n^2) donde se realizaban las comparaciones en el vector índice y luego se realizaban los cambios en ambos vectores manteniendo así la posición de la fecha y la línea de datos a la misma en la misma posición. Sin embargo esta función hizo que el programa fuera altamente ineficiente y era la de mayor complejidad, por lo que luego decidimos reemplazarla por una función que organizara los datos por el método de merge cuya complejidad es O(n log n), lo cual ayuda altamente a la eficiencia del programa y reducido drásticamente el tiempo que el programa tomaba para ejecutarse, posteriormente en main se realizaba un ciclo for donde se extraían los datos del vector con los datos de la bitácora y eran guardados de forma ya ordenada. Por último, una función realizaba la búsqueda tiene complejidad linear 0(n), pues el usuario ingresa los datos deseados y son transformados al formato de nuestro índice, para que luego sean imprimidos los datos que estén dentro del rango deseado.

En especialmente en el caso de la función ordenamiento se puede apreciar como el reducir la complejidad de O(n^2) a O(n log n) ayudo de manera bastante significativa a la eficiencia del programa, aunque el ciclo de merge sea mas extenso de programar, el hecho de que es así de eficiente lo hace una buena opción si lo que se busca es que el programa funcione mas rápido. En cuanto al algoritmo de búsqueda decidimos utilizar el índice previamente creado para facilitar la búsqueda en la búsqueda de los datos que se deseaban, tan solo haciendo comparaciones en un for con el valor de la fecha de cada dato.