

**Actividad 1.3**

**Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales**

Emma Fernanda Gómez Guerra (A00827097)

13 de septiembre de 2020

Para esta evidencia se utilizaron las metodologías y conceptos vistos en clase para poder realizar un programa que finalmente fue realizado por medio de una búsqueda binaria y un quicksort. Se buscó obtener un cierto rango de información que fuera otorgada por el usuario sobre un archivo "bitacora.txt". Fue importante primero ordenar este archivo para que el programa pudiera leerlo de la manera más conveniente y se pudiera ejecutar con éxito.

Cuando se está buscando algo dentro de un programa es esencial tener una lista de elementos donde se pueda obtener el elemento menor y el mayor, y a partir de éstos comenzar la búsqueda que se desee e ir actualizando el dato menor en cada paso. En clase se vieron los siguientes tipos de búsqueda y ordenamiento:

**Ordenamiento:**

- **Método de burbuja:** cuando se utiliza este método, lo que se hace es ir comparando los elementos adyacentes y empujando los valores menores hacia arriba (los mayores abajo).  
→ Complejidad:  $O(n^2)$
- **Método de selección:** Este método consiste principalmente de buscar el valor mínimo de la lista, intercambiar éste con el primero, buscar el que sigue a este valor mínimo en el resto de la lista, intercambiarlo con el segundo y así sucesivamente.  
→ Complejidad:  $O(n^2)$
- **Método merge:** Este método divide el problema en pequeños problema que a su vez se dividirán en más problemas, mientras no sean suficientemente pequeños.  
→ Complejidad:  $O(n \log n)$
- **Método de intercambio:** Es muy similar al método de burbuja, se comparan los elementos encontrando el menor y empujándolo hacia el inicio de la lista por intercambio.  
→ Complejidad:  $O(n^2)$
- **Método de inserción:** En este método se comienza con una lista ordenada. Teniendo  $k$  elementos ordenados de menor a mayor, se toma el elemento  $k+1$  y se compara con todos los que ya están ordenados, el programa se detiene cuando se encuentra un elemento menor o cuando ya no se encuentran elementos.  
→ Complejidad:  $O(n^2)$
- **Método de quicksort:** teniendo un arreglo de  $n$  elementos, se toma el valor del array como pivote, se separan los elementos menores a este a la izquierda y los mayores a la derecha. Se repite este proceso de forma recursiva hasta que estos tengan más de 1 elemento.  
Complejidad: Complejidad:  $O(n \log n)$

**Búsqueda:**

- **Búsqueda secuencial:** Este método va comparando el elemento que se busca con cada elemento del arreglo hasta cuando se encuentra. Es un método fácil de entender y usar pero no la más eficiente.
- **Búsqueda binaria:** Con este método los elementos pueden estar en cualquier orden. Y pueden llegar a hacerse  $n$  operaciones de comparación para poder obtener el resultado. La búsqueda binaria es el tipo de búsqueda más eficiente pues se puede hacer sobre un arreglo ordenado (como se hizo en la act 1.3 con el archivo 'bitácora.txt'). Su funcionamiento consiste en comparar si el valor buscado está en la mitad inicial o final. En la que esté, se divide en inicial y final de nuevo, y así hasta poder encontrar el valor que se está buscando.

En conclusión, es importante saber manejar y entender cada uno de estos métodos para poder crear programas de manera fácil y eficiente. Para la actividad 1.3, como mencioné anteriormente, utilicé la búsqueda binaria y el método de ordenamiento quicksort porque eran los más convenientes de usar ya que se tenía una lista desordenada y se buscaba encontrar un rango de información de esta misma. Con el quicksort se buscó ordenar la información del archivo y con la búsqueda binaria se buscaron los elementos ingresados por el usuario de esta lista. Además, fue nuevo pero fácil de usar struct en vez de clases pues simplificó mucho más el programa ya que no era necesario poner los métodos públicos y privados.

**Referencias (investigación):**

Alfaro Olave, T. (2020). Algoritmos de Búsqueda y Ordenamiento. Inf.utfsm.cl. Retrieved 15 September 2020, from <https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf>.