

Actividad 1.3: Reflexión individual

Estudiante: Diego Palma Rodríguez

Matrícula: A01759772

El ordenamiento y la búsqueda de datos son operaciones fundamentales en la computación, pues presentan un gran variedad de aplicaciones en diversas ramas de las ciencias matemáticas, computación, ingeniería e incluso hasta en el ámbito cotidiano. En la actualidad existen diversos algoritmos que nos permiten realizar dichas operaciones, y la selección y uso de los mismos depende de su eficiencia computacional (minimización de recursos como el tiempo, memoria, etc.). En la actividad 1.3 se aplicaron el algoritmo de ordenamiento por mezcla (*mergesort*) y el algoritmo de búsqueda binaria para la inspección de registros almacenados en un archivo de texto. A continuación, se realiza una reflexión acerca de su uso y eficiencia de estos algoritmos.

Los algoritmos de ordenamiento, como su nombre lo dice, nos permiten ordenar información en base a criterios como el orden temporal, numérico, alfabético, etc. El objetivo de ordenar un conjunto de datos, en esta actividad, es realizar una búsqueda más eficiente de información. Existen diversos tipos de algoritmos para lograr ello, como el ordenamiento por inserción, burbuja, mezcla, de ordenamiento rápido, etc. Por lo general, para seleccionar uno de dichos algoritmos nos basamos en su complejidad temporal expresado en la notación Big-O. En base a dicha notación se decidió utilizar el algoritmo de ordenamiento por mezcla (*mergesort*), pues presenta una complejidad $O(n\log(n))$ y, además, los elementos en el vector mantienen su posición original con respecto a los otros (estabilidad). Dado que se requiere ordenar una gran cantidad de datos este algoritmo es adecuado para la aplicación en cuestión, pues si utilizáramos otro algoritmo, como el de selección, podría tardar decenas de minutos en ordenar los datos de nuestros datos de prueba. Por este motivo es de gran relevancia seleccionar adecuadamente un algoritmo de ordenamiento.

Con respecto a los algoritmos de búsqueda, estos nos permiten encontrar un dato de interés dentro de un gran conjunto de datos. Por lo general, un algoritmo de búsqueda es más eficiente cuando se realiza sobre un conjunto de datos ordenados. Al igual que con los algoritmos de ordenamiento, la selección de estos tipos de algoritmos se basó en la complejidad temporal. En la actividad se nos pide implementar el algoritmo de búsqueda binaria que tiene un costo computacional $O(\log(n))$; sin embargo, existen otros métodos como la búsqueda utilizando una tabla o mapa HASH cuya complejidad es $O(1)$, lo cual nos permitiría encontrar datos al instante. Como se comentó en los algoritmos de ordenamiento, la importancia de la eficiencia temporal nos permite ahorrar una gran cantidad de tiempo en la ejecución de nuestro programa, de lo contrario el usuario podría llegar a esperar horas hasta tener una respuesta.

En conclusión, los algoritmos utilizados son lo suficientemente eficientes para resolver la actividad en un tiempo corto y adecuado. Si bien existen otros algoritmos que suelen ser más eficientes, para este caso en particular nos basta con los algoritmos implementados. En caso se tuviera que optar por ordenar una mayor cantidad de datos que los que se tienen como datos de prueba se debería optar por utilizar algoritmos más eficientes aún.