

Natalia Sofía Salgado García

A01571008

Reflexión

Para una actividad como esta, donde existe una cantidad grande de registros, lo más eficiente sería utilizar algoritmos de ordenamiento y de búsqueda con complejidad logarítmica. En cuanto al ordenamiento, los algoritmos de Bubble Sort, Insertion Sort y Selection Sort tienen una complejidad de $O(n^2)$ en el peor caso; mientras que MergeSort tiene una complejidad de $O(n \log n)$ en el peor caso. Hablando sobre QuickSort, éste algoritmo tiene una complejidad en el peor caso de $O(n^2)$, no obstante, en casos promedio tiene una complejidad de $O(n \log n)$. Entonces, lo más eficiente sería utilizar MergeSort o QuickSort. Adicionalmente, ya que los casos suelen ser casos promedios, el método QuickSort es aún más eficiente que el de MergeSort porque requiere menos memoria.

En cuanto a la búsqueda de elementos, se puede hacer una búsqueda secuencial que tiene una complejidad de $O(n)$ o una búsqueda binaria cuya complejidad es $O(\log n)$. Claramente, lo óptimo sería utilizar la búsqueda binaria puesto que requiere menos tiempo debido al aspecto logarítmico de éste.

LINK AL VIDEO:

<https://drive.google.com/file/d/1vzedl8U3429V5vYxJUy04zQz-oXF8TB6/view?usp=sharing>