

Evidencia 1 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Con respecto a los algoritmos de ordenamiento disponibles en C ++, no existe un algoritmo principal o mejor, todos se utilizan y funcionan mejor de acuerdo al caso, la cantidad, la forma en la que se presentan los datos, etc. Cada caso particular tendrá un algoritmo que funcione mejor. Para elegir el algoritmo que mejor se adapte a cada situación, se debe comprender a fondo el problema a resolver. Algunas preguntas clave para determinar cuál usar son:

- **¿Qué grado de orden tendrá la información a manejar?**

Si se trata de información que está o estará casi ordenada, se puede optar por un algoritmo sencillo como BubbleSort. Si van a estar desordenados, puede usarse Quicksort. Todo depende del caso.

- **¿Qué cantidad de datos vas a manipular?**

Si la cantidad de datos es pequeña, se puede optar por usar un algoritmo sencillo en lugar de un complejo.

- **¿Qué tipo de datos se van a ordenar?**

Algunos algoritmos solo funcionan con un tipo específico de dato y otros pueden manejar todos.

- **¿Qué tamaño tienen los registros de tu lista?**

Si los registros son muy grandes, algoritmos que realizan muchos intercambios pueden resultar lentos.

- Bubble Sort: Consiste en ciclar repetidamente a través de la lista, comparando elementos adyacentes de dos en dos. Si un elemento es mayor que el que está en la siguiente posición se intercambian
- Quick Sort: Utiliza recursividad para buscar posiciones en la lista y hace ordenamientos.

Fuentes:

C Con Clase | Ordenamiento (introduccion). (2022). Retrieved 12 September 2022, from <https://conclase.net/c/orden>