

Reflexión personal – Fin de Unidad Temática

De esta unidad, pienso que lo más importante fue entender y conocer cada algoritmo de ordenación para saber cuál es mejor en cada caso, ya que ninguno es el mejor siempre: cada uno se adapta mejor a distintos contextos. Me agradó ver también cómo funcionan internamente y poder programar la mayoría de ellos.

Los primeros que vimos fueron *insertion sort* y *shellsort*. El primero es muy intuitivo, ya que en cada iteración externa se inserta el elemento en su lugar final del arreglo. Es bueno si la lista ya está ordenada, pero sentí que era muy específico e ineficiente en general. Después de eso, vimos *shellsort*, que extiende la lógica del *insertion sort*, agrupando los elementos por incrementos y no comparando solo los adyacentes. Esto hace que no se desplace de uno en uno, y en ciertos casos donde podríamos mover un elemento mucho más hacia atrás, lo hacemos sin tantas iteraciones. Me agradó ver que el rendimiento con esta implementación mejora muchísimo.

Luego vimos los métodos de intercambio, como *bubble sort* y *quicksort*. *Bubble sort* me pareció bastante simple. Aun así, fue útil para entender cómo funcionan los intercambios. Por otro lado, *quicksort* fue el que más me costó entender. La idea de dividir en base a un elemento (pivote) y aplicar el algoritmo de forma recursiva a cada subparte me pareció bastante interesante. Lo que sí noté es que, en el código, si el método llama internamente a una recursión muy profunda, se puede generar un *stack overflow* por el límite de la pila.

Finalmente, a nivel práctico y teórico, vimos los algoritmos de selección, como *selection sort* y *heapsort*. *Selection sort* busca siempre el menor elemento en la parte no ordenada del arreglo y lo ubica en su lugar. No me pareció muy útil, la verdad, pero en cuanto a estructura de código, era bastante corto. En cambio, *heapsort* me pareció muy interesante. El uso de una estructura de árbol binario representado con un arreglo para ordenar elementos de forma subordinada fue un concepto que se me hizo raro en un principio. Pero al final entendí como construir un *heap* permite ir extrayendo los elementos de manera ordenada, sin necesidad cada par posible.

Me hubiera gustado ver otros algoritmos más a fondo, como *radix sort*, *bucket sort*, *merge sort*, *counting sort* y otros más, pero considero que en esta unidad aprendí lo fundamental.

En lo personal, me agradó mucho esta unidad, pero sentí que cada tema se vio muy rápido y no profundizamos en formas prácticas de implementar cada uno ni en los contextos específicos en los que conviene usarlos.