Trabajo Integrador Programación

Alumno:

Luis Enrique Denegri

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Fecha de Entrega:

9 de junio de 2025

Tabla de contenido

Introducción

En el campo de la programación y la informática, los algoritmos de búsqueda y ordenamiento son fundamentales para la manipulación eficiente de datos. Estos algoritmos permiten encontrar elementos dentro de estructuras de datos y organizar colecciones de elementos de manera ordenada, lo que optimiza el rendimiento de los programas y facilita otras operaciones como la búsqueda y el análisis de datos.

Algoritmos de Búsqueda

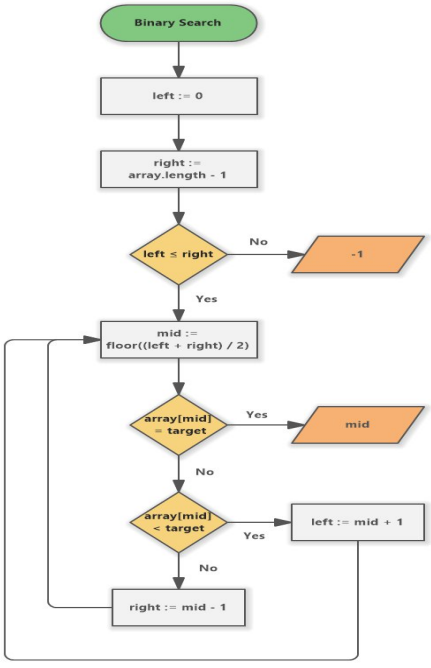
Los algoritmos de búsqueda son procedimientos diseñados para localizar un elemento dentro de una estructura de datos, como una lista o un árbol. Los más conocidos son:

1. Búsqueda Secuencial (Lineal)

Este algoritmo recorre la lista desde el inicio hasta el final, comparando cada elemento con el valor buscado. Su ventaja radica en su simplicidad y facilidad de implementación. Sin embargo, su eficiencia es lineal, es decir, su tiempo de ejecución crece proporcionalmente con la cantidad de elementos en la lista (Big O: O(n)).

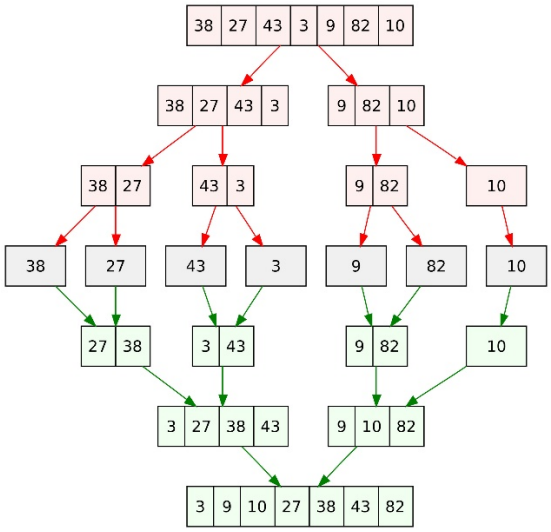
2. Búsqueda Binaria

Utiliza la estrategia de divide y vencerás para localizar un elemento en una lista ordenada. Se compara el elemento medio de la lista con el valor buscado y, según el resultado, se descarta la mitad de la lista en cada paso. Su eficiencia es logarítmica (Big O: O(log n)), lo que la convierte en una opción mucho más rápida que la búsqueda lineal en listas ordenadas.



Algoritmos de Ordenamiento

Los algoritmos de ordenamiento reordenan los elementos de una lista o estructura de datos según un criterio, como orden ascendente o descendente. Los algoritmos más relevantes incluyen:



- ✓ Bubble Sort

Compara pares de elementos adyacentes y los intercambia si están en el orden incorrecto. Este proceso se repite hasta que toda la lista está ordenada. Su complejidad temporal es $O(n^2)$, lo que lo hace ineficiente para grandes volúmenes de datos (Cormen et al., 2009).

- ✓ Insertion Sort

Construye la lista ordenada de manera incremental: inserta cada elemento en su posición correcta dentro de la lista parcialmente ordenada. Su eficiencia es $O(n^2)$ en el peor caso, aunque es eficiente para listas pequeñas o parcialmente ordenadas (Knuth, 1998).

- ✓ Selection Sort

Selecciona repetidamente el elemento más pequeño (o más grande) de la lista desordenada y lo coloca en la posición correcta. Aunque conceptualmente sencillo, su complejidad es también $O(n^2)$ (Sedgewick & Wayne, 2011).

- ✓ Merge Sort

Un algoritmo de ordenamiento eficiente que sigue el paradigma de divide y vencerás. Divide la lista en dos mitades, las ordena recursivamente y luego las fusiona. Su complejidad es O(n log n) (Cormen et al., 2009).

- ✓ Quick Sort

También utiliza divide y vencerás. Selecciona un elemento pivote y particiona la lista en dos

sublistas: una con elementos menores al pivote y otra con elementos mayores. Luego ordena las sublistas de forma recursiva. En promedio, su eficiencia es O(n log n) (Hoare, 1962).
Comparación de Algoritmos

Conclusión

El entendimiento y estudio de los algoritmos de búsqueda y ordenamiento son esenciales en la educación de programadores y científicos de datos, dado que su correcta aplicación no solo mejora el desempeño de los sistemas, sino que también asegura una experiencia de usuario más fluido y eficaz.

Referencias

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press. Link: https://mitpress.mit.edu/9780262033848/introduction-to-algorithms
- Knuth, D. E. (1998). *The Art of Computer Programming*, *Volume 3: Sorting and Searching* (2nd ed.). Addison-Wesley. Link: https://www.pearson.com/en-us/subject-catalog/p/art-of-computer-programming-volume-3-sorting-and-searching-the-2nd-edition/P200000001634/9780201896855
- Hoare, C. A. R. (1962). Quicksort. *The Computer Journal*, 5(1), 10–16. Link: https://doi.org/10.1093/comjnl/5.1.10
- Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley. Link: https://algs4.cs.princeton.edu/home/