

Entender el proceso de selección: en cada pasada, identificar el elemento más pequeño de la porción no ordenada de la lista y colocarlo en su posición correcta.

El algoritmo selecciona repetidamente el elemento más pequeño (o más grande) de la parte no ordenada de la lista y lo intercambia con el primer elemento de la parte no ordenada. Este proceso se repite para la parte restante sin ordenar hasta que se ordene toda la lista.

Implementar el algoritmo de Selection Sort en cpp.

<https://github.com/Marambulag/Algorithms>

Evaluar la eficiencia del algoritmo, discutiendo la complejidad temporal y espacial.

complejidad temporal: $O(N^2)$ a

complejidad espacial: $O(1)$

Comparar el rendimiento de Selection Sort con otros algoritmos de ordenamiento, como Bubble Sort, observando diferencias en tiempos de ejecución y número de comparaciones.

Advantages of Selection Sort Algorithm

Simple and easy to understand.

Works well with small datasets.

Disadvantages of the Selection Sort Algorithm

Selection sort has a time complexity of $O(n^2)$ in the worst and average case.

Does not work well on large datasets.

Does not preserve the relative order of items with equal keys, which means it is not stable.

Probar la implementación con diferentes conjuntos de datos para identificar cómo varía su comportamiento dependiendo del tamaño y el orden inicial de los elementos.

Debido a su complejidad entre más grande el array peor funcionaba

Referencia

<https://www.geeksforgeeks.org/selection-sort-algorithm-2/>