Ordenamiento por Selección

Pedro Gordillo

March 15, 2025

Índice

Ordenamiento por Selección

Definición

El ordenamiento por selección es un algoritmo de ordenamiento. Su funcionamiento consiste en dividir el arreglo en dos partes: la parte ordenada y la parte desordenada. En cada iteración, el elemento más pequeño de la parte desordenada se selecciona y se intercambia con el primer elemento de la parte desordenada, ampliando así la parte ordenada.

Usos y Aplicaciones

- Se utiliza en pequeñas listas donde la eficiencia no es un problema.
- Puede ser útil en sistemas con restricciones de memoria debido a su estabilidad en el uso de espacio.

Ventajas

- Fácil de implementar y comprender.
- No requiere memoria adicional, ya que es un algoritmo en sitio (in-place).

Desventajas

- Su eficiencia es baja en comparación con otros algoritmos como QuickSort o MergeSort.
- Requiere muchas comparaciones e intercambios, lo que lo hace ineficiente en grandes conjuntos de datos.

Paso a Paso del Algoritmo

El proceso del ordenamiento por seleccion se puede dividir en los siguientes pasos:

1. **Inicialización**
2. **División conceptual**
   - Una parte ordenada
   - Una parte desordenada
3. **Selección del minimo**
4. **Intercambio**
5. **Repetición**

Ejemplo con el Arreglo $\{87, 25, 1, 55, 9\}$

Paso a paso:

- Arreglo inicial: $\{87, 25, 1, 55, 9\}$
- **Paso 1**: Se busca el minimo en $\{87, 25, 1, 55, 9\}$ y es 1. Se intercambia con el primer elemento.

$$\{1, 25, 87, 55, 9\}$$

- **Paso 2**: Se busca el minimo en $\{25, 87, 55, 9\}$ y es 9. Se intercambia con el segundo elemento.

$$\{1, 9, 87, 55, 25\}$$

- **Paso 3**: Se busca el minimo en $\{87, 55, 25\}$ y es 25. Se intercambia con el tercer elemento.

$$\{1, 9, 25, 55, 87\}$$

- **Paso 4**: Se busca el minimo en $\{55, 87\}$ y es 55. No se necesita intercambiar.
- **Resultado**: $\{1, 9, 25, 55, 87\}$

Notación y complejidad Big-O

La notación Big O se usa para describir el rendimiento de los algoritmos en función del tamaño de la entrada.

Implementación en Java

```java
public class OrdenamientoPorSeleccion {
    public void ordenamientoSeleccion(int[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
            int menor = i;
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[j] < arr[menor]) {
                    menor = j;
                }
            }
            int temporal = arr[menor];
            arr[menor] = arr[i];
            arr[i] = temporal;
        }
    }
}
```

Complejidad Ordenamiento por selección

En el caso del algoritmo de ordenamiento por selección:

- El bucle externo recorre la lista n veces.
- El bucle interno recorre la parte no ordenada de la lista, que en el peor caso tiene aproximadamente n iteraciones en la primera pasada, $n-1$ en la segunda, y así sucesivamente.
- Esto da como resultado una complejidad de tiempo de ejecución de $O(n^2)$, ya que en total se realizan aproximadamente $\frac{n(n-1)}{2}$ comparaciones.