

1. Dado el pseudocódigo del método de ordenación por Selección Directa, analiza detalladamente el orden del tiempo de ejecución

Eficiencia del algoritmo

Aunque este método es eficiente en términos de simplicidad y fácil de entender, tiene una eficiencia en tiempo de $O(n^2)$ en todos los casos, lo que lo hace menos adecuado para grandes volúmenes de datos en comparación con algoritmos más avanzados como Quicksort o Mergesort que tienen mejor eficiencia promedio y en el mejor caso.

Proceso de Selección Directa Paso a Paso

Vamos a comenzar con el arreglo original: 256,458,365,298,43,648

Paso 1: Buscar el mínimo y colocarlo en la primera posición

- **Comparación 1:** Comparar 256 con 458 → 256 es menor.
 - **Comparación 2:** Comparar 256 con 365 → 256 es menor.
 - **Comparación 3:** Comparar 256 con 298 → 256 es menor.
 - **Comparación 4:** Comparar 256 con 43 → 43 es menor.
 - **Comparación 5:** Comparar 43 con 648 → 43 es menor.
- Movimiento:** Intercambiar 43 y 256 → Arreglo resultante: 43,458,365,298,256,648

Paso 2: Buscar el mínimo en el subconjunto restante (ignorando la primera posición)

- **Comparación 6:** Comparar 458 con 365 → 365 es menor.
 - **Comparación 7:** Comparar 365 con 298 → 298 es menor.
 - **Comparación 8:** Comparar 298 con 256 → 256 es menor.
- Movimiento:** Intercambiar 256 y 458 → Arreglo resultante: 43,256,365,298,458,648

Paso 3: Continuar con el siguiente subconjunto

- **Comparación 9:** Comparar 365 con 298 → 298 es menor.
- Movimiento:** Intercambiar 298 y 365 → Arreglo resultante: 43,256,298,365,458,648

Paso 4: Proceder con los últimos dos elementos

- No se necesita intercambio ya que 365 es menor que 458 y 648.

Paso 5: Última posición

- No se necesita comparación ni movimiento ya que 648 es el último y ya es el mayor.

Resultado Final

- El arreglo queda ordenado como 43,256,298,365,458,648

- **Detalle de Comparaciones y Movimientos**
- **Comparaciones:**
 - **Paso 1:** 5 comparaciones (256 vs 458, 365, 298, 43, 648)
 - **Paso 2:** 4 comparaciones (458 vs 365, 298, 256)
 - **Paso 3:** 3 comparaciones (365 vs 298, 256)
 - **Paso 4:** 2 comparaciones (365 vs 458)
 - **Paso 5:** No se necesitan comparaciones porque solo queda un elemento.
- **Total de Comparaciones:** $5+4+3+2=14$ + $4 + 3 + 2 = 14$ + $4+3+2=14$
- **Movimientos (Intercambios):**
 - **Paso 1:** 1 movimiento (43 intercambiado con 256)
 - **Paso 2:** 1 movimiento (256 intercambiado con 458)
 - **Paso 3:** 1 movimiento (298 intercambiado con 365)
 - **Paso 4 y 5:** No se realizan movimientos porque los elementos ya están en la posición correcta.
- **Total de Movimientos:** 3

Funcionamiento del Algoritmo de Selección Directa

1. Inicialización:

- El algoritmo comienza en el primer índice del arreglo y busca el elemento más pequeño en el arreglo completo.

2. Colocación del mínimo:

- Una vez encontrado el mínimo en todo el arreglo, lo intercambia con el elemento en el primer índice. Ahora, el elemento en el primer índice es el menor de todo el arreglo y se considera ordenado.

3. Reducción del problema:

- En el siguiente paso, el algoritmo ignora el primer elemento (ya ordenado) y repite el proceso para el subconjunto del arreglo que comienza desde el segundo índice hasta el final.

4. Repetición:

- Este proceso se repite, cada vez comenzando la búsqueda del mínimo desde el siguiente índice no ordenado y comparándolo solo con los elementos a su derecha. Por lo tanto, con cada paso, la parte del arreglo que necesita ser considerada se reduce en uno.

5. Terminación:

- El algoritmo se detiene cuando llega al penúltimo elemento del arreglo, porque el último elemento automáticamente será el más grande después de que todos los demás elementos hayan sido ordenados y colocados en su posición.

¿Cómo sabe que está ordenado?

El algoritmo de Selección Directa no necesita verificar explícitamente si el arreglo está ordenado porque sistemáticamente coloca el mínimo de los elementos restantes en la posición correcta en cada paso. Dado que el proceso asegura que cada elemento colocado al principio del arreglo es el menor de todos los elementos restantes, el ordenamiento es inherentemente construido paso a paso.

3. ¿cuál es la relación de las comparaciones y movimientos respecto al algoritmo de Quicksort visto anteriormente?

La elección entre Quicksort y Selección Directa puede depender del tamaño del conjunto de datos y de las características del entorno en que se utilicen (por ejemplo, memoria disponible, necesidad de estabilidad en el ordenamiento, etc.). Quicksort es superior en la mayoría de los casos prácticos debido a su eficiencia promedio y al manejo eficiente de listas grandes, mientras que Selección Directa puede ser adecuado para conjuntos de datos pequeños o para propósitos educativos donde la simplicidad del algoritmo es una ventaja.