Selection Sort

Friday, December 26, 2025 2:21 AM

What is it?

A simple comparision-based sorting Algorithm.

Concept:

It divides the input list into two parts:

- a sorted sublist (built up from left to right)
- an unsorted sublist (the rest of the items).

Mechanism:

It repeatedly selects the smallest (minimum) element from the unsorted sublist and moves it to the end of the sorted sublist.

How it Works:

1. Set Boundary: start with the entire list considered "unsorted". The sorted portion is empty.
2. Find minimum: Scan the unsorted part of the list to find the element with the lowest value.
3. Swap: Swap this minimum element with the very first element of the unsorted part.
4. Advance: move the boundary of the sorted partition one step to the right.
5. Repeat: Repeat steps 2-4 for the remaining unsorted elements until the entire list is sorted.

Algorithm Complexity (Big O)

- Time Complexity:
  - Best Case: $O(n^2)$
  - Worst Case: $O(n^2)$
  - Average Case: $O(n^2)$

  Note: Unlike Bubble Sort, Selection Sort always scans the remaining list to find the minimum, even if the list is already sorted.
- Space Complexity: $O(1)$

  It is an in-place algorithm.

Key characteristics

- Swaps: Performs min. no. of swaps (at most n swaps). This is useful when writing to memory is expensive.
- Stability: Unstable

  It might change the relative order of equal elements (e.g., swapping a "5" from the start to the middle might skip over another "5").
- Performance: Generally performs better than Bubble sort due to fewer swaps, but still inefficient for large lists.

Pseudocode

```
FUNCTION SelectionSort(list)
    n = length(list)
    FOR i FROM 0 TO n-1
    /* Assume the current position holds the min */
      minIndex = i
      /* check the rest of the array for a smaller value */
      FOR j FROM i+1 TO n-1
        IF list[j] < list[minIndex] THEN
            minIndex = j
        END IF
      END FOR
      /* Swap the found min. with the current position */
      IF minIndex != i THEN
        Swap(list[i], list[minIndex])
      END IF
    END FOR
END FUNCTION.
```