Bubble Sort

Thursday, December 25, 2025 7:22 PM

What is it?

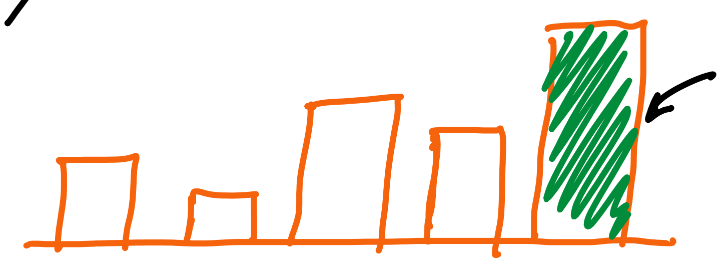
A simple, comparision-based sorting algorithm.

Concept:

It repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order.

Naming:

It is called "Bubble sort" because with each iteration, the largest element "bubbles up" to its correct position at the end of the list (like a bubble rising in water.).



largest element to its correct position at the end of the list.

How it works

1. Start: Begin at index 0.
2. Compare: Compare the current element (arr[i]) with the next element (arr[i+1])
3. Swap: if $\boxed{arr[i] > arr[i+1]}$, swap their position.
4. Iterate: Continue this comparision/swap pair until the end of the array is reached. At the end of Pass 1, the largest item is guarnteed to be in the last spot.
5. Repeat: Start over from index 0 for the remaining unsorted elements (ignoring the last "sorted" spots).
6. Stop: The algorithm stops when a pass completes with zero swaps (meaning the list is sorted).

Algorithm Complexity

- Time Complexity
  - Best Case: O(n)

    The array is already sorted (requires an "optimized" flag to detect no swaps).
  - Worst Case: $O(n^2)$

    The array is reverse sorted.

- Space Complexity: O(1)

  It is an in-place sorting algorithm (requires very little extra memory).

Key Characteristics

- Efficiency: Very slow and inefficient for large datasets. Primarily used for educational purposes.
- Stability: Stable.

  It preserves the relative order of equal elements (eg: If there are two 5s, the one that came first stays first).
- Simplicity: Easiest sorting logic to implement.