

- Definition: A simple, comparison-based sorting algorithm.
- Core Concept: Repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order.
- Analogy: The largest unsorted element "bubbles" up to its correct position during each pass.
- Algorithm Steps:
  1. Start a loop from the beginning of the list to the second-to-last element. This is the "pass" loop.
  2. Inside this loop, start another loop to compare adjacent elements.
  3. Compare the current element with the next element.
  4. If the current element is greater than the next element, swap them.
  5. After the first pass, the largest element is at the end of the list.
  6. Reduce the range of the next pass to exclude the already sorted elements at the end.
  7. Repeat the passes until the list is sorted.
- Time Complexity (Exam Point):
  - Worst Case:  $O(n^2)$  - when the list is in reverse order.
  - Average Case:  $O(n^2)$
  - Best Case:  $O(n)$  - when the list is already sorted (only if an optimization is used).
- Space Complexity (Exam Point):  $O(1)$  - It is an in-place sorting algorithm.
- Key Characteristics:
  - Stable: The relative order of equal elements is preserved.
  - In-place: Requires minimal additional memory space.
- Optimization:
  - Introduce a boolean flag (e.g., `swapped`).
  - If a full pass is completed with no swaps, the list is sorted, and the algorithm can terminate early. This is crucial for achieving  $O(n)$  best-case performance.
- Advantages:
  - Simple to understand and implement.
  - Good for introducing the concept of sorting algorithms.
  - Efficient for very small or nearly-sorted lists.
- Disadvantages:
  - Very inefficient for large datasets.
  - Outperformed by algorithms like Insertion Sort, Merge Sort, and Quick Sort in most scenarios.
- Common Exam Questions:
  - Trace the state of a given array after each pass of bubble sort.
  - Calculate the total number of swaps or comparisons for a given list.
  - Explain the worst-case scenario and why it results in  $O(n^2)$  complexity.
  - Compare the performance of bubble sort with selection sort or insertion sort.