Bubble Sort and Insertion Sort

Ali Akbari

June 2025

Understanding Time Complexity

- **Big O (O)**: Upper bound of algorithm runtime (worst case).
- **Big Omega (Ω)**: Lower bound (best case).
- **Big Theta (Θ)**: Tight bound (average case).
- Example: Sorting a list of n items may take $O(n^2)$ in the worst case (e.g., Bubble Sort).

Bubble Sort Example

- **Problem**: Arrange 5 students by height: [160, 175, 155, 180, 165] cm.
- Compare adjacent pairs, swap if out of order (taller first).
- **Step-by-Step**:
  1. Pass 1: [160, 175, 155, 180, 165] $\rightarrow$ [160, 155, 175, 165, 180]
  2. Pass 2: [155, 160, 175, 165, 180] $\rightarrow$ [155, 160, 165, 175, 180]
  3. Pass 3: No swaps needed, sorted: [155, 160, 165, 175, 180]

Bubble Sort

- Repeatedly compare adjacent elements and swap if in wrong order.
- Each pass “bubbles” largest unsorted element to the end.
- **Time Complexity**:
  - Worst/Average: $O(n^2)$ (many comparisons/swaps).
  - Best: $O(n)$ (already sorted, with optimization).
- **Space Complexity**: $O(1)$ (in-place).

Insertion Sort Example

- **Problem**: Sort cards: [5, 2, 8, 1].
- Insert each card into its correct position in the sorted portion.
- **Step-by-Step**:
  1. Start: [5], rest: [2, 8, 1]
  2. Insert 2: [2, 5, 8, 1]
  3. Insert 8: [2, 5, 8, 1]
  4. Insert 1: [1, 2, 5, 8]

Insertion Sort

- Build sorted portion by inserting each element into its correct position.
- Shift larger elements right to make space.
- **Time Complexity**:
  - Worst/Average: $O(n^2)$ (shifting elements).
  - Best: $O(n)$ (nearly sorted).
- **Space Complexity**: $O(1)$ (in-place).