Merge Sort and Quick Sort

Ali Akbari

June 2025

Merge Sort Example

- **Problem**: Sort exam scores: [65, 90, 50, 80, 70, 95, 60, 85].
- Split into halves, sort, and merge by comparing elements.
- **Step-by-Step**:
  1. Split: [65, 90, 50, 80], [70, 95, 60, 85]
  2. Split: [65, 90], [50, 80], [70, 95], [60, 85]
  3. Merge: [50, 65, 80, 90], [60, 70, 85, 95]
  4. Final: [50, 60, 65, 70, 80, 85, 90, 95]

Merge Sort

- Divide list into halves, recursively sort, then merge sorted halves.
- Merge compares smallest elements from each half.
- **Time Complexity**: $O(n \log n)$ (always divides and merges).
- **Space Complexity**: $O(n)$ (temporary arrays for merging).

Quick Sort Example

- **Problem**: Sort cards: [5, 2, 8, 1, 9, 3], pivot = last (3).
- Partition around pivot, recurse on sublists.
- **Step-by-Step**:
  1. Partition: [2, 1, 3, 5, 8, 9]
  2. Left [2, 1]: [1, 2]
  3. Right [5, 8, 9]: [5, 8, 9]
  4. Final: [1, 2, 3, 5, 8, 9]

Quick Sort

- Choose pivot, partition list (smaller left, larger right), recurse.
- Pivot choice affects performance (e.g., last element).
- **Time Complexity**:
  - Worst: $O(n^2)$ (sorted list).
  - Average/Best: $O(n \log n)$.
- **Space Complexity**: $O(\log n)$ (recursion stack).