

Divide and Conquer Algorithms

Divide and Conquer Algorithms

1. Definition

- A problem-solving strategy that breaks a problem into smaller subproblems of the same type, solves them recursively, and combines their solutions.

2. Three Steps

- **Divide**: split the problem into subproblems.
- **Conquer**: solve subproblems recursively (base case solved directly).
- **Combine**: merge subproblem solutions into the final answer.

3. Classic Examples

- **Merge Sort**: divides the array in half, sorts each half, then merges - $O(n \log n)$ time.
- **Quick Sort**: partitions around a pivot, then recursively sorts each partition.
- **Binary Search**: repeatedly halves the search space.

4. Analyzing Divide and Conquer

- Runtime is often expressed using recurrence relations (e.g., $T(n) = 2T(n/2) + O(n)$), solved using the Master Theorem.

5. Why It's Powerful

- Breaking large problems into smaller ones often yields significantly better time complexity than brute-force approaches (e.g., $O(n \log n)$ vs. $O(n^2)$ sorting).