Data Structures and Algorithms

Lecture 04

Aniket Basu Roy

2026-01-19 Mon

Contents

1 Agenda

1.1 Merge Sort

- Proof of Correctness
- Time Complexity

2 Merge Sort

2.1 Divide-Conquer-Combine Approach

- Divide the input into a number of subproblems.

- Conquer the subproblems by solving them recursively.
- Combine the solved subproblems to return the solution to the original problem.

3 Merge Sort

3.1 Input: $A[1:n]$

3.2 Output: $A[1:n]$ is sorted.

- Divide $A[1:n]$ into $A[1:\lfloor n/2 \rfloor]$ and $A[\lfloor n/2 \rfloor + 1:n]$
- Recursively run Merge Sort on $A[1:\lfloor n/2 \rfloor]$ and $A[\lfloor n/2 \rfloor + 1:n]$
- Merge the two sorted arrays to sort $A[1:n]$

3.3 An Example:

3.4 Description of the Algorithm

3.5 Proof of Correctness

- Inductive Proof
- Loop Invariant in the Merge function
- Draw the flow chart

3.6 Time Complexity

- Recurrence Relation

$$T(1) = \Theta(1)$$
$$T(n) = 2T(n/2) + \Theta(n)$$

- Prove using Recursion Tree

$$T(n) = \Theta(n \log n)$$