Data Structures and Algorithms

Lecture 05

Aniket Basu Roy

2026-01-21 Wed

Contents

1 Agenda

1.1 Recurrence Relation (using Asymptotic Notation)

1.2 Merge Sort – Time Complexity

1.3 Recursion Tree

$$T(1) = \Theta(1)$$

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

- Prove using Recursion Tree

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

1.4 Substitution Method

- Prove for $T(n) = 2T(n/2) + n$
- Guess for T(n) and prove by induction

1.4.1 Try doing using asymptotic notation

$$T(1) = \Theta(1)$$

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

- Express using constants while proving using induction
  - Separately for the upper bound
  - Separately for the lower bound

1.4.2 Pitfalls to avoid

- Avoid asymptotic notation in the inductive hypothesis of the substitution method.

1.4.3 Excercise

$T(n) = T(n/3) + T(2n/3) + \Theta(n)$

- Guess the upper and lower bounds using the recursion tree
- Substitute the guessed function in the recurrence relation