Data Structures and Algorithms

Lecture 05

Aniket Basu Roy

BITS Pilani Goa Campus

2026-01-21 Wed

Agenda

- Recurrence Relation (using Asymptotic Notation)
- Merge Sort – Time Complexity
- Recursion Tree
- Substitution Method
- The Master Theorem

Merge Sort

```
MERGE-SORT(A, p, r)
1  if p ≥ r                          // zero or one element?
2      return
3  q = ⌊(p + r)/2⌋                   // midpoint of A[p : r]
4  MERGE-SORT(A, p, q)               // recursively sort A[p : q]
5  MERGE-SORT(A, q + 1, r)           // recursively sort A[q + 1 : r]
6  // Merge A[p : q] and A[q + 1 : r] into A[p : r].
7  MERGE(A, p, q, r)
```

Merge Sort

Time Complexity

- Recurrence Relation

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

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

Merge Sort

Recursion Tree

Substitution Method

Excercise

$$T(n) = 4T(n/4) + n$$

Excercise

$$T(n) = 4T(n/4) + n$$

Guess

$$T(n) = O(n), \text{or } \exists c > 0, T(n) \leq cn$$

Substitution

$$T(n) \leq 4(cn/4) + n = cn + n = O(n)$$

Excercise

$$T(n) = 4T(n/4) + n$$

Guess

$$T(n) = O(n), \text{or } \exists c > 0, T(n) \leq cn$$

Substitution

$$T(n) \leq 4(cn/4) + n = cn + n = O(n)$$

WRONG!

Excercise

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