# Data Structures and Algorithms

Lecture 06

Aniket Basu Roy

2026-01-23 Fri

## Contents



# 1 Agenda

## 1.1 Recurrence Relation (continued from the previous lecture)

### 1.1.1 Excercise 1

$$T(n) = 8T(n/2) + \Theta(n^2)$$

- Guess 1: $T(n) \leq dn^3$
- Guess 2: $T(n) \leq dn^3 - d'n^2$

### 1.1.2 Excercise 2

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

- Guess the upper and lower bounds using the recursion tree

- Substitute the guessed function in the recurrence relation
- What happens if we take the complete binary tree on the height of the tree?
  - How many leaves are there and where is the function dominated more?

## 1.2 The Master Theorem

$$T(n) = aT(n/b) + f(n)$$

$a \geq 1,\ b > 1,\ f(n) \geq 0$

### 1.2.1 Case 1

If $\exists \varepsilon > 0,\ f(n) = O(n^{\log_b a - \varepsilon})$, then $T(n) = \Theta(n^{\log_b a})$.

### 1.2.2 Case 2

If $\exists k \geq 0,\ f(n) = \Theta(n^{\log_b a} \log^k n)$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.

### 1.2.3 Case 3

If $\exists \varepsilon > 0, f(n) = \Omega(n^{\log_b a + \varepsilon})$ and $\exists c < 1, n_0 > 0, \forall n \geq n_0, \quad af(n/b) \leq cf(n)$, then $T(n) = \Theta(f(n))$.