

Recurrence Relation: $T(n) = T(n-1) + n$ (Decreasing Function)

This video covers analyzing a recursive function with a decreasing argument, deriving its recurrence relation, and solving it using two methods: **Recursion Tree** and **Substitution (Back Substitution)**.

1. The Recursive Function & Time Analysis

Function structure:

When n is passed:

- Calls itself with $n-1$: $T(n-1)$
- Contains a loop that runs **n times** (for loop with $n+1$ iterations, but asymptotically n)

Time breakdown per call:

- Condition check: **1 unit**
- Loop body (executes n times): **n units**
- Recursive call: **$T(n-1)$ units**

Exact recurrence: $T(n) = T(n-1) + 2n + 2$

Simplified (Big O notation): $T(n) = T(n-1) + n$ (linear class, degree 1)

Base case: $T(0) = 1$ (constant time, does nothing)

2. Method 1: Recursion Tree Method

Visualize the recursion calls:

Tree expansion:

- Level 0: n
- Level 1: $n-1$

- Level 2: $n-2$
-...
- Level $n-1$: 1
- Level n : $T(0) = 1$

Total time = Sum of all levels:

$$T(n) = n + (n-1) + (n-2) + \dots + 1 + T(0)$$

Formula: $T(n) = \frac{n(n+1)}{2}$

Asymptotic: $\Theta(n^2)$

3. Method 2: Substitution (Back Substitution) Method

Start with: $T(n) = T(n-1) + n$

Step 1: First substitution

$$T(n-1) = T(n-2) + (n-1)$$

$$\rightarrow T(n) = [T(n-2) + (n-1)] + n$$

$$\rightarrow T(n) = T(n-2) + (n-1) + n$$

Step 2: Second substitution

$$T(n-2) = T(n-3) + (n-2)$$

$$\rightarrow T(n) = [T(n-3) + (n-2)] + (n-1) + n$$

$$\rightarrow T(n) = T(n-3) + (n-2) + (n-1) + n$$

Step 3: Generalize to k steps

$$T(n) = T(n-k) + (n-k+1) + (n-k+2) + \dots + (n-1) + n$$

Step 4: Base case ($k = n$)

$$n-k = 0 \rightarrow T(n) = T(0) + 1 + 2 + \dots + (n-1) + n$$

$T(0) = 1$, so:

$$T(n) = 1 + \sum_{i=1}^n i = 1 + \frac{n(n+1)}{2}$$

Asymptotic: $\Theta(n^2)$

4. Key Patterns & Exam Tips

Exam Type	Approach	Time
Theory/Essay	Show complete tree OR full substitution steps	Detailed
Objective/MCQ	Direct: "Sum = $n(n+1)/2 \rightarrow \Theta(n^2)$ "	Quick

Practice pattern recognition:

- $T(n) = T(n-1) + f(n)$ where $f(n) = \text{linear} \rightarrow \Theta(n^2)$
- Loop runs n times + 1 recursive call = $n + T(n-1)$