



Design and Analysis of Algorithms

Vandana M L

Department of Computer Science & Engineering

DESIGN AND ANALYSIS OF ALGORITHMS

Solving Recurrences

Slides courtesy of **Anany Levitin**

Vandana M L

Department of Computer Science & Engineering

$$T(n) = T(n-1) + 1 \quad n > 0 \quad T(0) = 1$$

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

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

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

...

$$= T(n-i) + i$$

...

$$= T(n-n) + n = n = O(n)$$

Design and Analysis of Algorithms

Solving Recurrences: Example2



$$\begin{aligned}T(n) &= T(n-1) + 2n - 1 & T(0) &= 0 \\&= [T(n-2) + 2(n-1) - 1] + 2n - 1 \\&= T(n-2) + 2(n-1) + 2n - 2 \\&= [T(n-3) + 2(n-2) - 1] + 2(n-1) + 2n - 2 \\&= T(n-3) + 2(n-2) + 2(n-1) + 2n - 3\end{aligned}$$

...

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

...

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

$$= 0 + 2 + 4 + \dots + 2n - n$$

$$= 2 + 4 + \dots + 2n - n$$

$$= 2 * n * (n+1) / 2 - n$$

// arithmetic progression formula $1 + \dots + n = n(n+1)/2$ //

$$= O(n^2)$$

$$T(n) = T(n/2) + 1 \quad n > 1$$

$$T(1) = 1$$

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

$$= T(n/2^2) + 1 + 1$$

$$= T(n/2^3) + 1 + 1 + 1$$

.....

$$= T(n/2^i) + i$$

.....

$$= T(n/2^k) + k \quad (k = \log n)$$

$$= 1 + \log n$$

$$= O(\log n)$$

$$T(n) = 2T(n/2) + cn \quad n > 1 \quad T(1) = c$$

$$T(n) = 2T(n/2) + cn$$

$$= 2(2T(n/2^2) + c(n/2)) + cn = 2^2 T(n/2^2) + cn + cn$$

$$= 2^2 (2T(n/2^3) + c(n/2^2)) + cn + cn = 2^3 T(n/2^3) + 3cn$$

.....

$$= 2^i T(n/2^i) + icn$$

.....

$$= 2^k T(n/2^k) + kcn \quad (k = \log n)$$

$$= nT(1) + cn \log n = cn + cn \log n$$

$$= O(n \log n)$$



THANK YOU

Vandana M L

Department of Computer Science & Engineering

vandanamd@pes.edu