# 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

## Solving Recurrences: Example1

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

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

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

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

…

$= T(n-i) + i$

…

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

## Solving Recurrences: Example2

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

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

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

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

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

...

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

...

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

$= 0 + 2 + 4 +...+ 2n - n$

$= 2 + 4 +...+ 2n - n$

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

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

$= O(n^2)$

## Solving Recurrences: Example3

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

$$T(n) = T(n/2) + 1$$
$$= T(n/2^2) + 1 + 1$$
$$= T(n/2^3) + 1 + 1 + 1$$
$$\ldots\ldots$$
$$= T(n/2^i) + i$$
$$\ldots\ldots$$
$$= T(n/2^k) + k \quad (k = \log n)$$
$$= 1 + \log n$$
$$= O(\log n)$$

## Solving Recurrences: Example4

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

$$
\begin{aligned}
T(n) &= 2T(n/2) + cn \\
&= 2(2T(n/2^2) + c(n/2)) + cn = 2^2T(n/2^2) + cn + cn \\
&= 2^2(2T(n/2^3) + c(n/2^2)) + cn + cn = 2^3T(n/2^3) + 3cn \\
&\ldots\ldots \\
&= 2^iT(n/2^i) + icn \\
&\ldots\ldots \\
&= 2^kT(n/2^k) + kcn \quad (k = \log n) \\
&= nT(1) + cn\log n = cn + cn\log n \\
&= O(n\log n)
\end{aligned}
$$

# THANK YOU

**Vandana M L**
Department of Computer Science & Engineering

**vandanamd@pes.edu**