Recursive Equations

Recursive Equations

Analysis of recursive algorithms results in recursive equations. If $T(n)$ is the time taken to find the max element in an array of size n, then.

- $T(n) = c_1$, if $n = 1$
- $T(n) = T(n-1) + c_2$, otherwise

The closed-form solution is $\cdots$

Recursive Equations

Analysis of recursive algorithms results in recursive equations. If $T(n)$ is the time taken to find the max element in an array of size n, then.

- $T(n) = c_1$, if $n = 1$
- $T(n) = T(n-1) + c_2$, otherwise

The closed-form solution is $\cdots$

$T(n) = c_2.(n-1) + c_1$

Recursive Equations

- At times recursive equations can be tricky.
- **Master Theorem** is used to solve recurrence equations in asymptotic terms.

Recursive Equations

Recurrence relations can also be used to define a sequence

- Example: $x_{n+1} = c.x_n$, given $(n \geq 0; x_0 = 1)$

The closed-form solution is ...

Recursive Equations

Recurrence relations can also be used to define a sequence

- Example: $x_{n+1} = c.x_n$, given $(n \geq 0; x_0 = 1)$

The closed-form solution is ...
$x_n = c^n$

Recursive Equations

A slightly different one: $x_{n+1} = b_{n+1}.x_n$

Given:

- $n \geq 0$
- the value of x_0
- the set $\{b_1, b_2, \cdots\}$

The closed-form solution is ...

Recursive Equations

A slightly different one: $x_{n+1} = b_{n+1}.x_n$

Given:

- $n \geq 0$
- the value of x_0
- the set $\{b_1, b_2, \cdots\}$

The closed-form solution is ...
$x_n = b_n.b_{n-1}.b_{n-2} \cdots b_1.x_0$

Recursive Equations

Let us raise the ante further: $x_{n+1} = b_{n+1}.x_n + c_{n+1}$

Given:

- $n \geq 0$
- the value of x_0
- the set $\{b_1, b_2, \cdots\}$
- the set $\{c_1, c_2, \cdots\}$

Recursive Equations

Let us raise the ante further: $x_{n+1} = b_{n+1}.x_n + c_{n+1}$

Given:

- $n \geq 0$
- the value of x_0
- the set $\{b_1, b_2, \cdots\}$
- the set $\{c_1, c_2, \cdots\}$

Hint: Reduce it to the form $y_{n+1} = y_n + d_{n+1}$

Example: $x_{n+1} = 3.x_n + n$, where $(n \geq 0; x_0 = 0)$

If $x_n = 3^n.y_n$ then we get

$y_{n+1} = y_n + n/3^{n+1}$, where $n \geq 0; y_0 = 0$

Example: $x_{n+1} = 3.x_n + n$, where $(n \geq 0; x_0 = 0)$

If $x_n = 3^n.y_n$ then we get

$$y_{n+1} = y_n + n/3^{n+1}, \text{ where } n \geq 0; y_0 = 0$$

Finally we get: $x_n = 3^n \sum_{j=1}^{n-1} j/3^{j+1}$

Recursive Equations

Till now we have only considered first-order recursive equations.

Note:

- In first-order recursive equations, the current value depends only on the previous value.
- In second-order recursive equations, the current value depends on the previous **two** values.

Recursive Equations

Let the equation be: $x_{n+1} = a.x_n + b.x_{n-1}$

Given:

- $n \geq 1$
- the value of x_0, x_1 and a, b

Recursive Equations

Let the equation be: $x_{n+1} = a.x_n + b.x_{n-1}$

Given:

- $n \geq 1$
- the value of x_0, x_1 and a, b

Hint: Call for a trial solution (as you solve second-order differential equations)

Recursive Equations

Let the equation be: $x_{n+1} = a.x_n + b.x_{n-1}$

Follow the following steps:

- Let the trial solution be $x_n = \alpha^n$, and substitute it in the given equation
- We obtain the quadratic equation $\alpha^2 = a.\alpha + b$
- If α_+ and α_- are the distinct roots, the general solution is $x_n = c_1.\alpha_+^n + c_2.\alpha_-^n$
- The constants c_1 and c_2 will be determined so that x_0, x_1 have the assigned values.

Example

$L_0 = 100000, L_1 = 200000$, and $L_n = (L_{n-1} + L_{n-2})/2$

- The characteristic polynomial is $\cdots$
- The roots are $\cdots$
- The general solution is $\cdots$
- The values of c_1 and c_2 are $\cdots$

Example

$L_0 = 100000, L_1 = 200000$, and $L_n = (L_{n-1} + L_{n-2})/2$

- The characteristic polynomial is $x^2 - x/2 - 1/2$
- The roots are 1 and $-1/2$
- The general solution is $L_n = c_1 + c_2(-1/2)^n$
- $c_1 = 500000/3$ and $c_2 = -200000/3$.