CS412: Solving Recurrences

In Engineering Mathematics, you learned to solve ODEs in Calculus and their different types (First-Order/higher, linear/non-linear, homogeneous/non-homogeneous) with Constant/variable coefficients. All are useful when it comes to solving recurrences and analyzing algorithms running time represented as recurrent relations.

Recurrences are the discrete counterparts of differential equations and are called difference equations

In Algorithm Analysis, most recursive algorithms can be represented by a recurrence and the time/space complexity is the solution of the recurrence.

A Simple example: Factorial $n!$

A recursive version would be

Factorial (n)

1. if $n = 1$ return 1
2. return $n * \text{Factorial}(n-1)$

Let $T(n)$ be the time to compute $n!$

Then,

$$\underbrace{T(n) = n \cdot T(n-1)}_{\text{recurrence}}, \quad n \geq 1, \quad \underbrace{T(1) = 1}_{\text{base-case}}$$

Solving

$$T(n) = n \cdot T(n-1)$$

$$= n \cdot [(n-1) \cdot \underbrace{T(n-2)}]$$

$T(n-1)$: time to solve for $n-1$

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

$$\vdots$$

$$= n(n-1) \cdot (n-2) \ldots \underbrace{T(1)}_{=1} \Rightarrow \boxed{T(n) = n!}$$

or $\boxed{T(n) = \Theta(n!)}$

The Fibonacci Sequence

Input: $n \geq 1$ (ie, $n \in \mathbb{Z}^+$)

Output: the n^{th} Fibonacci number.

Psuedocode. Fib(n):

1. If $n = 0$ or $n = 1$, then return n
2. return $Fib(n-1) + Fib(n-2)$

Let,

$F(n)$ be the time to compute the n^{th} Fibonacci number.

then,

$$\underbrace{F(n) = F(n-1) + F(n-2)}_{\substack{2^{nd} \text{ order recurrence} \\ /\text{difference equation}}} \quad \begin{cases} \text{two boundary conditions.} \\ F(1) = F(0) = 1. \end{cases}$$

Let us wait for the above and consider a slightly different equation

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

or make it simpler,

$$F(n) = F(n-1) + F(n-1) + 1, \quad F(0) = 0$$

or

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

Let's get the first few terms to build our intuition

$F(0) = 0$ (Given)

$F(1) = 2\underbrace{F(0)}_{=0} + 1 = 1$

$F(2) = 2\underbrace{F(1)}_{=1} + 1 = 3$

$F(3) = 2\underbrace{F(3-1)}_{F(2)=3} + 1 \quad$ RC $= 2(3) + 1 = 7$

and so on.

It is clear from the pattern that the n^{th} term is

$$F(n) = 2^n - 1, \ \forall n \geq 0$$

Let's Unroll the recurrence

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

$$= 2\left[2F(n-2) + 1\right] + 1$$

$$= 2^2 F(n-2) + 2 + 1 = 2^2\left[2F(n-3)+1\right] + 2 + 1$$

$$= 2^3 F(n-3) + 2^2 + \underbrace{2^1}_{=2} + \underbrace{2^0}_{=1}$$

$$\vdots$$

ie, $F(n) = \underbrace{2^n F(n-n)}_{=0} + 2^{n-1} + 2^{n-2} + \dots + 2^0$

$$= 2^n \underbrace{F(0)}_{=0} + 2^{n-1} + 2^{n-2} + \dots + 2^0$$

$\therefore$ $F(n) = \sum_{j=1}^{n-1} 2^j = 2^n - 1$

$\therefore$ $\boxed{F(n) = \Theta(2^n)}$

Let us generalize the above form:

$$F(n) = 2F(n-1) + 1, \quad \underbrace{F(0) = 0}_{\text{base-case}}$$

$\underbrace{A(n) = b\,A(n-1) + f(n)}_{\text{Generic form.}}$; boundary condition.

RC

Date

ie,

$$A(n) = b\,A(n-1) + f(n)$$

$$= b\left[b\,A(n-2) + f(n) \right] + f(n)$$

$$= b^2\,A(n-2) + b^1 . f(n) + b^0 f(n)$$

$$= b^2\left[b\,A(n-3) + f(n) \right] + b^1 f(n) + b^0 f(n)$$

$$= b^3\,A(n-3) + b^2 . f(n) + b^1 f(n) + b^0 f(n)$$

$$\vdots$$

$$\Rightarrow A(n) = b^n \underbrace{A(n-n)}_{=c,\ \text{some constant}} + b^{n-1} f(n) + b^{n-2} f(n) + \cdots + b^0 f(n)$$

$$\boxed{\therefore A(n) = b^n . c + \sum_{j=1}^{n-1} b^j f(n)}$$

Generic form of a solution for $A(n)$

DIFFERENCE EQUATIONS

FIRST- ORDER RECURRENCES

First order recurrences such that where the function on n is expressed in terms of the function evaluated at $(n-1)$.

eg.

$F(n) = 2F(n-1)+1$, is a first-order recurrence

Recall, the Fibonacci recurrence:

$F(n) = F(n-1) + F(n-2)$ is a second-order recurrence

Recurrence relations are commonly called as Difference Equations (the discrete counterparts of Differential Equations)

So, we have a discrete operator ∇ (difference operator)

$$\nabla A_n = A_n - A_{n-1}$$

The techniques to solve difference equations are similar to what you studied in Engineering Mathematics.

LINEAR HOMOGENEOUS RECURRENCES WITH CONSTANT COEFFICIENTS

Consider a relation of the form:

$$a_0 A_n + a_1 A_{n-1} + \cdots + a_i A_{n-i+1} + a_i A_{n-i} = 0 \quad \text{—— (1)}$$

where,

a_i's are all constant

Equation is a linear, homogeneous recurrence, with constant coefficients (a_i) or order i.

Guessing the solution of eq '1' has the form

$$A_n = r^n \quad \left[\text{Recall, } F(n) = 2F(n-1)+1\right]$$

where $r > 0$

RC

Date

Replacing $A_\alpha = r^\alpha$ in eq (1), we get

$$a_0 r^n + a_1 r^{n-1} + a_2 r^{n-2} + \dots + a_i r^{n-i} = 0$$

or

$$r^{n-i}(a_0 r^i + a_1 r^{i-1} + \dots + a_{i-1} r + a_i) = 0$$

Dividing both sides by r^{i-1} [$\because r > 0$]

we get,

$$\boxed{a_0 r^i + a_1 r^{i-1} + \dots + a_{i-1} r + a_i = 0} \quad (2)$$

also known as the Characteristic Equation

and the L.H.S. is called the characteristic polynomial of order i

Suppose that the characteristic equation (2) has solutions $r_1, r_2, \dots, r_i$ then, if all solutions are distinct,

the most general form of the solution of the given recurrence is:

$$A_n = \underbrace{c_1 r_1^n + c_2 r_2^n + \dots + c_i r_i^n}_{\text{a linear combination}}$$

where, [Recall from EM!]

c_i's are constants and can be determined using boundary conditions

<u>For example:</u>

Given a <u>second-order</u> <u>linear</u> <u>homogeneous</u> <u>recurrence</u> with <u>constant</u> coefficients.

$$\underbrace{A_n - 3A_{n-1} + 2A_{n-2}}_{\text{order-2}} = 0 \quad \text{homogeneous}$$

So,

$$A_n - 3A_{n-1} + 2A_{n-2} = 0$$

Let,

$A_n = r^n$ be our <u>guess of the solution</u>

So,

the above <u>equation</u> becomes

$$r^n - 3r^{n-1} + 2r^{n-2} = 0$$

or

$$r^{n-2}(r^2 - 3r + 2) = 0$$

or $\quad r^2 - 3r + 2 = 0 \quad [\because r > 0]$

or

$$(r-1)(r-2) = 0$$

or

$$r = \{1, 2\}$$

The general solution of the given recurrence is

$A_n = c_1 r_1^n + c_2 r_2^n$, c_1 & c_2 are constants and $n > 0$

$$A_n = c_1(1^n) + c_2(2^n)$$

or

$$\boxed{A_n = n(2^n)}$$

RC

Now, let's return to Fibonacci:

Recall, $F(n) = F(n-1) + F(n-2)$

or, using a more convenient notation:

$$F_n = F_{n-1} + F_{n-2}$$

or $\quad F_n - F_{n-1} - F_{n-2} = 0$ [A linear, homogeneous difference equation of order 2, with constant coefficients]

Guess the solution of F_n to be r^n,

we can write,

$$r^n - r^{n-1} - r^{n-2} = 0$$

or $\quad r^{n-2}(r^2 - r - 1) = 0$,

or $\quad r^2 - r - 1 = 0 \quad [\because r > 0]$

We'll solve the above characteristic equation using the Quadratic formula:

[Recall, $ax^2 + bx + c = 0$ then, $x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$]

ie,

$$r = \frac{-(-1) \pm \sqrt{(-1)^2 - 4(1)(-1)}}{2(1)}$$

or $\quad r = \dfrac{1 \pm \sqrt{5}}{2}$, the two roots are $r_1 = \underbrace{\dfrac{1+\sqrt{5}}{2}}_{\text{Golden Ratio!}}$; $r_2 = \dfrac{1-\sqrt{5}}{2}$

Let r_1 be ϕ and r_2 be $\hat{\phi}$

Then, the general solution of the recurrence (Fibonacci)

$$F(n) = \alpha_1 r_1^n + \alpha_2 r_2^n$$

or $\quad F(n) = \alpha_1 \phi^n + \alpha_2 \hat{\phi}^n$, where α_1 & α_2 are constants

We can solve for α_1 & α_2 using the boundary conditions:

$$F(0): \alpha_1 \cdot \phi^0 + \alpha_2 \hat{\phi}^0 = 0 \qquad [F(0) = 0,$$
$$F(1): \alpha_1 \phi^1 + \alpha_2 \hat{\phi}^1 = 1 \qquad F(1) = 1]$$

So, $\alpha_1 = \dfrac{1}{\sqrt{5}}$ and $\alpha_2 = -\dfrac{1}{\sqrt{5}}$

Hence, the **complete** solution of $F(n) = F(n-1) + F(n-2)$ with $F(0) = 0$ and $F(1) = 1$

is given as

$$\boxed{F(n) = \frac{1}{\sqrt{5}}\left(\phi^n - \hat{\phi}^n\right)}$$

Practise problems :

Solve :

1) $A(n) = 1.5\,A(n-1) + 1$, $A(0) = 0$

2) $B(n) = 2B(n-1) + \frac{n}{2}$ with $B(0) = 1$

3) $C(n) = C(n-1) - C(n-2)$, with $C(0) = 0$, $C(1) = 1$

4) $\alpha(n) = 2\alpha(n-1) + \alpha(n-2)$, $\alpha(0) = 0$, $\alpha(1) = 2$

5) $X_{n+1} = 2X_n - X_{n-1}$, with $X(0) = 0$ and $X(1) = 1$

x———————x

We've seen Unrolling and solution of linear recurrences as ways to solve recurrences :

Other methods include :

1) The Substitution Method (uses mathematical induction)
2) The Recurrence Tree Method
3) The Master Theorem
4) Generating Functions et al.