Outline # 5

Recurrence Relation:

A recurrence relation for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely $a_0, a_1, \ldots, a_{n-1}$ for all integers n with $n \geq n_0$ where n_0 is a non-negative integer.

A sequence is called a solution of a recursively ~~define a sequence~~ relation if its terms satisfy the recurrence relation.

Ex Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1, 2, 3, \ldots$ and suppose that $a_0 = 2$. What are a_1, a_2 & a_3?

We see from the recurrence relation that

$a_1 = a_0 + 3$
$a_1 = 2 + 3 = 5$

$a_2 = a_1 + 3$
$a_2 = 5 + 3 = 8$

$a_3 = a_2 + 3$
$a_3 = 8 + 3 = 11.$

Scanned with CamScanner

Solving Linear Recurrence Relations

A Linear homogenous recurrence relation of degree k with constant coefficients in a recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$$

where

$c_1, c_2, \dots, c_k$ are real numbers, and $c_k \neq 0$

=> In recurrence relation the right side is the sum of previous terms of sequence each multiplied by a function of n.

Solving Linear Homogenous Recurrence Relations with Constant Coefficient

The basic method for solving linear homogenous recurrence relation is to look for solutions of the form $a_n = r^n$ where r is a constant.

∴ $a_n = r^n$ is the solution of recurence relation if and only if

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \dots + c_k r^{n-k}$$

Scanned with CamScanner

For two distinct roots

Let c_1 and c_2 be real numbers. Suppose ther $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2. Then the sequence $\{a_n\}$ is a solution of the recurrence relation

$$a_n = c_1 a_{n-1} + \cancel{c_2} \; c_2 a_{n-2}$$

if and only if

$$\boxed{a_n = \alpha_1 r_1^n + \alpha_2 r_2^n}$$

for $n = 0, 1, 2, \ldots$

where α_1 & α_2 are constant

Example:

What is the solution of the recurrence relation

$$a_n = a_{n-1} + 2a_{n-2}$$

with $a_0 = 2$ & $a_1 = 7$?

Solution

$\because a_n = c_1 a_{n-1} + c_2 a_{n-2}$

or

$a_n = a_{n-1} + 2a_{n-2}$

so $c_1 = 1$ & $c_2 = 2$

The characteristic equation of this recurrence relation will be

$$r^2 - r - 2 = 0$$

$\because$ $\cancel{r^2 = c_1 r}$

$\therefore$ $r^2 - c_1 r_1 - c_2 = 0$

Its roots will be

$$r^2 - 2r + r - 2 = 0$$

$$r(r-2) + 1(r-2)$$

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

$r - 2 = 0$ $\quad$ $r + 1 = 0$

$\boxed{r_1 = 2}$ $\quad$ $\boxed{r_2 = -1}$

Hence the sequence $\{a_n\}$ is solution of recurrence relation if

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

$r_1 = 2, \; r_2 = -1$

$$\Rightarrow a_n = \alpha_1 2^n + \alpha_2 (-1)^n \text{ — (A)}$$

For $a_0 = 2$

$$a_0 = \alpha_1 (2)^0 + \alpha_2 (-1)^0$$

$$2 = \alpha_1 + \alpha_2 \text{ — (i)}$$

For $a_1 = 7$

$$a_1 = \alpha_1 (2)^1 + \alpha_2 (-1)^1$$

$$7 = 2\alpha_1 - \alpha_2 \text{ — (ii)}$$

Solving eq (i) & (ii)

$$2 = \alpha_1 + \cancel{\alpha_2}$$
$$7 = 2\alpha_1 - \cancel{\alpha_2}$$

$$9 = 3\alpha_1$$

$$\boxed{\alpha_1 = 3}$$

putting value of α_1 in eq (i)

$$2 = \alpha_1 + \alpha_2$$
$$2 = 3 + \alpha_2$$
$$-1 = \alpha_2$$
$$\boxed{\alpha_2 = -1}$$

putting values in eq (A)

$$a_n = 3(2)^n + (-1)\cdot(-1)^n$$

$$\Rightarrow a_n = 3(2)^n - (-1)^n$$

Example

page 518
example #5

Divide & Conquer Algorithm

Many recursive algorithms take a problem with a given input and divide it into one or more smaller problems. This reduction is successively applied until the solutions of the smaller problems can be found quickly. Such procedure is known as divide and conquer, and are called divide and conquer algorithms.

For Example:

we perform binary search by reducing the search for an element in the list to the search for this element in a list half as long. We successively apply this reduction until one element is left.

Divide & Conquer Recurrence Relations

Suppose that a recurresive algorithm divides a problem of size 'n' into 'a' subproblem, where each subproblem is of size n/b ('n' is multiple of 'b'). Also suppose that g(n) are extra operations are required conquer steps.

Then f(n) represent no of operations required to solve the problem of size 'n'.

$$\Rightarrow f(n) = a f(n/b) + g(n)$$

This is divide-and conqure recurrence relation.

Use of Divide-and-Conquer Recurresison relation in some important algorithms

(i) Binary Search:

The binary search algorithm reduces the search for an element in a search sequence of size n to the binary search for this element in search of size n/2, where n is even.
Hence if f(n) is the number of comparision required to search for an element in a search sequence of size 'n', then

$$f(n) = f(n/2) + 2$$

$\because$ n is even.

(ii) Merge Sort:

The merge sort algo splits a list to be sorted with n items, into two lists with n/2 elements each, and uses fewer than n comparisions to merge the two sorted list of n/2 items each into one sorted list

$$M(n) = 2M(n/2) + 2$$