Department of Distance Education

Punjabi University, Patiala

Class : B.A. 2 (Mathematics) **Semester : 4**
Paper : Opt.III (Discrete Mathematics-II) **Unit : I**
Medium : English

Lesson No.

SECTION-A

Department website : www.pbidde.org

B.A. PART-III | **MATHEMATICS OPT.-III**
SEMESTER-VI | **DISCRETE MATHEMATICS-II**

LESSON NO. 1.1 | **Author: Dr. Chanchal**

ANALYSIS OF ALGORITHMS

Structure:

1.1.0 Objectives

The prime goal of this unit is to enlighten the basic concepts of algorithm study, recurrence relations, discrete numeric functions and generating functions. During the study in this particular lesson, our main objective is to discuss problems that can be solved by using step-by-step methods, more formally known as algorithms. Further, we have discussed in detail about the

- Characteristics of algorithms.
- Efficiency of algorithms.
- Growth rates: the O notation.

1.1.1 Introduction to Algorithm

The word algorithm comes from the name of Persian author, Abu Jafar, who wrote a book on mathematics. It has several applications and the work regarding algorithm has gained significant importance. In computer science, the **analysis of algorithms** is the determination of the amount of resources (such as time and storage) necessary to execute them or we can say that algorithms are used to design a method that can be used by the computer to find out the solution of a particular problem. Most algorithms are designed to work with inputs of arbitrary length. Usually, the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity).

Algorithm analysis is an important part of a broader computational complexity theory, which provides theoretical estimates for the resources needed by any algorithm which solves a given computational problem. These estimates provide an insight into reasonable directions of search for efficient algorithms. So, an algorithm may be defined as follows :

- An algorithm is a set of rules for carrying out calculation either by hand or on a machine.
- It is a finite step-by-step list of well-defined instructions for solving a particular problem.
- An algorithm is a sequence of computational steps that transform the input into the output.

For example : The algorithm described below is designed to find out the minimum of three numbers a, b and c.

1. min= a
2. If b < min, then min= b .
3. If c < min, then min= c .

1.1.2 Various Characteristics of Algorithms

1. **Input :** The algorithm starts with an input or we can say that the algorithm receives input. The input involves the supply of one or more quantities.
2. **Output :** The algorithm ends with an output or we can say that the algorithm produces output. The result we obtain at the end is called output and at least one quantity is produced.
3. **Precision :** The steps involved in the algorithm are precisely stated. Each instruction mentioned in the algorithm should be clear and unambiguous.
4. **Determinism :** The intermediate results of each step of execution are unique and determined only by the inputs and the results of the preceding steps.
5. **Finiteness :** The number of steps in an algorithm should be finite. It means that if we trace out the instruction of an algorithm, then for all the cases, the algorithm must terminate after a finite number of steps.
6. **Correctness :** The output produced by an algorithm must be correct.
7. **Generality :** The algorithm must apply to a set of inputs.
8. **Effectiveness :** Every instruction stated in an algorithm should be very basic and clear so that it can be carried out very effectively.

1.1.3 Study of Algorithms

The study of algorithm involves several important and active areas of research out of which the most essential are discussed below:

1. **Creating an Algorithm :** The art of creating an algorithm can never be fully automated. By mastering these design strategies, it will become very easy to design new and useful algorithms.
2. **Algorithm Validation :** After the process of design of an algorithm, it is necessary to check that it computes the correct answer for all the possible legal inputs. This process is known as algorithm validation.
3. **Analysis of Algorithm or Performance Analysis :** As an algorithm is executed, it uses the computer's central processing unit (CPU) to perform operations and its memory to hold the program and data. Analysis of algorithms refers to the task of determining the computing time and storage that an algorithm requires and it should be done with great mathematical skills.
4. **Testing a Program :** It consists of two phases: debugging and profiling (or performance measurement). Debugging is the process of executing programs on sample data set to investigate whether faulty errors occur and, if so, correct them. Profiling is the process of executing a correct program on data sets and measuring the time and space it takes to compute the results.

1.1.3.1 Aspects of Algorithm Efficiency

The two important aspects of algorithm efficiency are:

I. The amount of time required to execute an algorithm and

II. The amount of memory space needed to run a program.

A computer requires a certain amount of time to carry out arithmetic operations. Moreover, different algorithms need different amount of space to hold numbers in memory for later use. An analysis of the time required to execute an algorithm of a particular size is referred to as the time complexity of the algorithm while an analysis of the computer memory required involves the space complexity of the algorithm.

Let M be an algorithm and n be the size of the input data. The time and space used by the algorithm are the two main features for the efficiency of M. The time is measured by counting the number of key operations. **For example :** In sorting and searching, one counts the number of comparisons but in arithmetic, one counts multiplications and neglects the additions.

These key operations are so defined that the time for the other operations is more than or at most proportional to the time for the key operations. The space is measured by counting the maximum of memory needed by an algorithm.

1.1.3.2 Some Important Functions

Functions play an important role in the study of algorithms and their analysis. Some of the important mathematical functions which are used very often in algorithms, are discussed below:

1. **Absolute Value Function :** Let x be any real number. Then, the absolute value of x, denoted by $|x|$ may be defined as

   $$|x| = \begin{cases} x, x \geq 0 \\ -x, x < 0 \end{cases}$$

   For example : $|-2| = 2, |9| = 9$.

2. **Characteristic Function :** Let A be any set and S be any subset of A. Then, characteristic function denoted by $c_S = \begin{cases} 1, x \in S \\ 0, x \notin S \end{cases}$

   For example : If $A = \{a, b, c\}$ and $S = \{a, b\}$, then $c_S(a) = 1, c_S(b) = 1, c_s(c) = 0$ because $a, b \in S$ and $c \notin S$. So, we can write

   $c_S = \{(a,1), (b,1), (c,0)\}$.

3. **Floor Function :** For any real number x, the floor function of x means the greatest integer which is less than or equal to x. It is denoted by $\lfloor x \rfloor$.

   For example : $\lfloor 2.58 \rfloor = 2, \lfloor -4.4 \rfloor = -5, \lfloor 2 \rfloor = 2$.

4. **Ceiling Function :** For any real number x, the ceiling function of x means the least integer which is greater than or equal to x. It is denoted by $\lceil x \rceil$.

   For example : $\lceil 2.58 \rceil = 3, \lceil -4.4 \rceil = -4, \lfloor 2 \rfloor = 2$.

5. **Integer Function :** For any real number x, the integer function of x converts x into an integer by deleting the fractional part of x. It is denoted by INT(x).

   For example : INT(2.44)=2, INT(-4.44)=-4.

   Note : (i) If x is an integer, then $\lfloor x \rfloor = \lceil x \rceil$. Otherwise $\lfloor x \rfloor + 1 = \lceil x \rceil$.

   (ii) $\lfloor x \rfloor = n \Rightarrow n \leq x < n+1$ and $\lceil x \rceil = n \Rightarrow n-1 < x \leq n$.

   (iii) INT(x)=$\lfloor x \rfloor$ if x is positive and INT(x)=$\lceil x \rceil$ if x is negative.

6. **Remainder Function :** Let M be a positive integer and k be any integer. Then, k (mod M) is called the remainder function and it denotes the integer remainder when k is divided by M. Also, k (mod M) is a unique integer such that $k = Mq + r$ where $0 \leq r < M$.

Note : (i) For positive numbers, we simply divide k by M to obtain remainder r but for negative numbers, we divibe $|k|$ by M to get remainder r' and $k\,(\text{mod } M) = M - r'$ if $r' \neq 0$.

For example : 26(mod4)=2 and -35(mod9)=9-8=1.

7. Logarithm and Exponent Functions : Let b be any positive integer. The logarithm of any positive number x to base b is written as $\log_b x$ and it represent exponent to which b must be raised to obtain x. Mathematically, we can write $y = \log_b x$ iff $b^y = x$.

For example : $\log_3 216 = 6$

Note : (i) For any base b, $\log_b 1 = 0$ and $\log_b b = 1$ because $b^0 = 1$ and $b^1 = b$.

(ii) Logarithm of a negative number and logarithm of zero is not defined.

1.1.4 Recursive Algorithm

A recursive algorithm is an algorithm which is used with smaller or simpler input values and which obtains the result for the current input by applying simple operations to the returned value for the smaller or simpler input. In other words, if a problem can be solved utilizing solutions to smaller versions of the same problem, and the smaller versions reduce to easily solvable cases, then one can use a recursive algorithm to solve that problem. For example, the elements of a recursively defined ser or a recursively defined function can be obtained by a recursive algorithm.

If a set or a function is defined recursively, then a recursive algorithm to compute its members or values describes the definition. Initial steps of the recursive algorithm correspond to the basis clause of the recursive definition and they identify the basis elements. It is then followed by the steps corresponding to the inductive clause, which reduce the computation for an element of one generation to that of elements of the immediately preceding generation.

1.1.5 Complexity of Algorithms

For an algorithm M, the complexity may be described by the function $f(n)$ which gives the running tine and/or storage space requirement of the algorithm in terms of the size n of the input data. In most of the cases, the storage space required by an algorithm is simply a multiple of the data size. Accordingly, unless otherwise stated or implied, the term complexity shall refer to the running time of the algorithm. The complexity function $f(n)$, which we assume gives the running time of an algorithm, usually depends not only on the size n of the input data but also on the particular data. In the complexity theory, the following two cases are usually investigated:

1. Worst Case : The maximum value of $f(n)$ for any possible input.
2. Average Case : The expected value of $f(n)$.

The analysis of average case assumes a use of probability distribution for the input data and we assume that the possible permutations of a data set are equally likely. Also, the following result is used for an average case:

Suppose the numbers $n_1, n_2, ..., n_k$ occur with respective probabilities $p_1, p_2, ..., p_k$, then the expectation or average value E is given by

$E = n_1 p_1 + n_2 p_2 + ... + n_k p_k$.

1.1.5.1 Standard Functions Measuring Complexity of Algorithms

Algorithms are generally compared or analysed on the basis of their complexity which is further measured in terms of the size of input data n described by the mathematical function $f(n)$. As n grows, complexity of algorithm M also increases and our interest is to measure this rate of growth. For the purpose, we compare $f(n)$ with some standard functions with different rate growths such that $\log_2 n, n, n\log_2 n, n^2, n^3, 2^n$. Now, complexity of any algorithm is measured in terms of these standard functions and we use a special notation for this, called Big-O notation, as defined below:

Big- O **:** Let $f(x)$ and $g(x)$ are functions defined on the set (or subset) of real numbers. Then, $f(x)$ is called order of $g(x)$ or big- O of $g(x)$, written as $f(x) = \mathrm{O}(g(x))$, if there exist a real number m and a positive constant c such that for all $x \geq m$, we have $|f(x)| \leq c|g(x)|$.

To show that $f(x) = \mathrm{O}(g(x))$ we have to find the value of m and c. Further, big-O gives an upper bound on number of key operations or we can say that big-O gives information about maximum number of key operations. For getting lower bound, we define the function big-omega (Ω).

Big-omega : Let $f(x)$ and $g(x)$ are functions defined on the set (or subset) of real numbers. Then, $f(x) = \Omega(g(x))$ of there exist positive constants c and k such that $|f(x)| \geq c|g(x)|$ for all $x \geq k$. Further, $f(x)$ is called big-omega of $g(x)$.

Big-theta : Let $f(x)$ and $g(x)$ are functions defined on the set (or subset) of real numbers. Then, $f(x) = \Theta(g(x))$ of there exist positive constants c_1, c_2 and k such that $c_1|g(x)| \leq |f(x)| \leq c_2|g(x)|$ for all $x \geq k$. Further, $f(x)$ is called big-theta of $g(x)$ if $f(x)$ is both big-O and big-omega of $g(x)$.

1.1.6 Growth Rate Functions

The time efficiency of almost all the algorithms can be characterized by the following growth rate functions:

1. **O(l)-Constant Time :** This means that the algorithm requires the same fixed number of steps regardless of the size of the task.
   For Example (Assuming a reasonable implementation of the task) :
   i. Push and pop operations for a stack (containing n elements);
   ii. Insert and remove operations for a queue.
2. **O(n)-Linear Time :** This means that the algorithm requires a number of steps proportional to the size of the task.
   For Example (Assuming a reasonable implementation of the task) :
   i. Traversal of a tree with n nodes;
   ii. Calculating n-factorial or n^{th} Fibonacci number by using the method of iteration.
3. **O(n^2)-Quadratic Time :** This means that the algorithm requires a number of steps proportional to the square of size of the task.
   For Example :
   i. Comparing two dimensional array of size n by n;
   ii. Find duplicates in an unsorted list of n elements (implemented with two nested loops).
4. **O($\log n$)-Logarithmic Time :**
   For Example :
   i. Binary search in a sorted list of n elements;
   ii. Insert and find operations for a binary search tree with n nodes.
5. **O($n \log n$)-"$n \log n$" Time :**
   For Example :
   i. More advanced sorting algorithms - quicksort, mergesort.
6. **O(a^n)($a > 1$) -Exponential Time :**
   For Example :
   i. Recursive Fibonacci Implementation;
   ii. Generating all permutations of n symbols.

Remarks : (i) The order of asymptotic behavior of the above described functions is

O(l)<O($\log n$)<O(n)<O($n \log n$)<O(n^2)-O(n^3)<O(a^n)

So, the best time is the constant time and the worst time is the exponential time and polynomial growth is considered manageable as compared to exponential growth.

(ii) If a function (which describes the order of growth of an algorithm) is a sum of several terms, its order of growth is determined by the **fastest growing term**. In particular, if we have a polynomial of the form

$p(n) = a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0$, then its growth is of the order n^k i.e., $p(n) = O(n^k)$.

1.1.7 Some Important Examples

Example 1.1 : Find $\lfloor \log_2 100 \rfloor$

Sol. $\because 2^6 = 64$ and $2^7 = 128$.

so $6 < \log_2 100 < 7$

$\Rightarrow \log_2 100 = 6.$

Example 1.2 : Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$.

Sol. Let $x \geq 1$

which gives $1 \leq x \leq x^2$ ....(1)

Now, $|f(x)| = |x^2 + 2x + 1| \leq |x^2| + |2x| + |1|$ [since $|x + y| \leq |x| + |y|$]

$\Rightarrow |f(x)| = x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$ [using (1)]

$\Rightarrow |f(x)| \leq 4|x^2| \forall x \geq 1$

which gives $f(x) = O(x^2)$

Example 1.3 : Suppose the polynomial $P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$ is of degree n. Show that $P(x) = O(x^n)$.

Sol. Let $x \geq 1$

so, $1 \leq x \leq x^2 \leq x^3 \leq .... \leq x^n$ ....(1)

Now, $|P(x)| = |a_0 + a_1 x + a_2 x^2 + ... + a_n x^n|$

$\leq |a_0| + |a_1 x| + |a_2 x^2| + ... + |a_n x^n|$ [since $|x + y| \leq |x| + |y|$]

$= |a_0| + |a_1| x + |a_2| x^2 + ... + |a_n| x^n$

$= |a_0| .1 + |a_1| x + |a_2| x^2 + ... + |a_n| x^n$

$\leq |a_0| x^n + |a_1| x^n + |a_2| x^n + ... + |a_n| x^n$ [using (1)]

$= [|a_0| + |a_1| + |a_2| + ... + |a_n|] x^n = cn^m$

where $c = |a_0| + |a_1| + |a_2| + ... + |a_n|$

so, $|P(x)| \leq c |x^n| \forall x \geq 1$

which gives $P(x) = O(x^n)$.

Example 1.4 : Find Big-O notation for $\log \angle n$. Further give Big-O estimate for $f(n) = 3n\log\angle n + (n^2+3)\log n$

Sol. Let n be any natural number.

As we know $\angle n = 1.2.3............n$

Now, $1 < 2 < 3 < 4...................... \le n$

so, $|\angle n| = 1.2.3.................n \le n.n.n...............n = n^n$

$\Rightarrow |\angle n| \le 1.n^n$ for all $n \ge 1$

so, $\angle n = O(n^n)$ with $c = 1, m = 1$

$\Rightarrow \log\angle n = O(\log n^n) = O(n\log n)$ ...(1)

For the second part, Let $n \ge 1$

so, $1 \le n \le n^2$ ...(2)

so, $|f(n)| = |3n\log\angle n + (n^2+3)\log n|$

$\le |3n\log\angle n| + |(n^2+3)\log n|$ [since $|x+y| \le |x| + |y|$]

$\le 3n.n\log n + (n^2 + 3.n^2)\log n$ [using (1) and (2)]

$= 3n^2\log n + 4n^2\log n$

$\Rightarrow |f(n)| \le 7.|n^2\log n|$ for all $n \ge 1$

so, f(n) is $O(n^2\log n)$ with $c = 7, m = 1$.

Example 1.5 : Prove that $f(x) = 8x^3 + 5x^2 + 7$ is $\Omega(g(x))$ where $g(x) = x^3$.

Sol. Let $x \ge 0$

then, $x^2 \ge 0$

Now, $|f(x)| = |8x^3 + 5x^2 + 7| \ge 8x^3$ $[\because 5x^2 + 7 \ge 0]$

$\Rightarrow |f(x)| \ge 8.|x^3|$ $\forall x \ge 0$

$\Rightarrow |f(x)| \ge 8.|g(x)|$ $\forall x \ge 0$

Hence $f(x)$ is $\Omega(g(x))$ where $c = 8, k = 0$.

1.1.8 Summary

In this lesson, we have studied about the algorithms. From our study, we can say that an algorithm is a sequence of instructions. Each individual instruction must be carried out, in its proper place, by the person or machine for whom the algorithm is intended. Consequently, an algorithm should always be considered in the context of certain assumptions. In more detail, we have discussed about the efficiency and complexity of algorithms, on the basis of which, we have learnt the procedure to compare the algorithms. Further, we have also given an idea of recursive algorithms that may be useful for understanding the recurrence relations which will be discussed in the next part of this unit.

1.1.9 Self Check Exercise

1) Show that $7x^2 - 9x + 4 = O(x^2)$.
2) Let $U = \{a, b, c, ..., x, y, z\}$ and $A = \{a, e, i, o, u\}$. Find the characteristic function of A.
3) Show that $g(n) = n^2(7n - 2)$ is $O(n^3)$.
4) Show that $x^4 + 9x^3 + 4x + 7$ is $O(x^4)$.
5) Find Big-O notation for $\angle n$.

1.1.10 Suggested Readings

1. Norman L. Biggs, *Discrete Mathematics*, Oxford University Press.
2. Harmohan Sharma, Ganesh Kumar Sethi, *Discrete Mathematics,* Sharma Publications, Jalandhar.
3. C.L. Liu, Elements of Discrete Mathematics (Second Edition), McGraw Hill, International Edition, Computer Science Series, 1986.

LESSON NO. 1.2 **Author: Dr. Chanchal**

DISCRETE NUMERIC FUNCTIONS AND GENERATING FUNCTIONS

Structure:

1.2.0 Objectives

The prime goal of this lesson is to enlighten the basic concepts of discrete numeric functions along with the detail elaboration of operations on numeric functions. Further, the knowledge about generating functions and several important results concerning them is also provided under this lesson.

1.2.1 Introduction

From our previous study, we are already familiar with the concept of function or mapping which may be defined as a rule $f: X \to Y$ that associates each element of X with a unique element of Y. Further, we recall the definition of sequence:

Def : Sequence

Let $\mathrm{N} = \{1,2,3,.....\}$ is the set of natural numbers and $\mathrm{Z} = \{..........,-3,-2,-1,0,1,2,3,..........\}$ is the set of integers. A mapping $S : \mathrm{N} \to \mathrm{Z}$ is called the sequence of integers. The image of any natural number n is called the n^{th} term of S and denoted by $S(n)$ or S_n. Further, n is also called the index or argument.

So, we can write $S = \{a_1, a_2, a_3, ..., a_n, ...\}$.

Note: (i) Numeric functions, discrete functions etc. are also used for sequence.

(ii) The sequences can also be expressed in a compact form known as the **closed form expression**. For example, the closed form expression of $\sum_{i=1}^{n} i = 1+2+....+n$ is $\frac{n(n+1)}{2}$.

1.2.2 Discrete Numeric Function

A function whose domain is the set of natural numbers and range is the set of real numbers is known as discrete numeric function or simply numeric function. If $a : \mathrm{N} \to \Re$ is a discrete numeric function, then $a(0)$=value of a at 0 is denoted by a_0. Similarly, $a(1) = a_1$, $a(2) = a_2$,............., $a(r) = a_r$,.........so on. Here, a_r represents the general form of numeric function $a = \{a_0, a_1, a_2, ...a_r, ...\}$.

1.2.3 Operations on Numeric Functions

Let a and b be two numeric functions and α be any real number. Then, we may define the following operations on numeric functions:

- **I.** **Sum:** The sum $a+b$ is a numeric function such that the value of $a+b$ at r is equal to the sum of the values of a and b at r.
- **II.** **Product:** The product $a.b$ is a numeric function such that the value of $a.b$ at r is equal to the product of the values of a and b at r.
- **III.** **Convolution:** The convolution of a and b denoted by $a*b$ is a numeric function c such that

$$c_r = a_o b_r + a_1 b_{r-1} + ... + a_r b_o = \sum_{i=0}^{r} a_i b_{r-i}$$

- **IV.** **Modulus of Numeric Function:** The modulus of a numeric function a may be defined as : $|a| = a_r$ if a_r is non-negative and $|a| = -a_r$ if a_r is negative.
- **V.** **Multiplication of Numeric Function by a Real Number:** The multiplication a numeric function a with real number α denoted by αa is also a numeric function whose value at r is equal to α times a_r.
- **VI.** **Forward Difference of a Numeric Function:** The forward difference of a numeric function a is also a numeric function whose value at r is equal to $a_{r+1} - a_r$. It is denoted by Δa.
- **VII.** **Backward Difference of a Numeric Function:** The forward difference of numeric function a is also a numeric function, denoted by ∇a , such that $\nabla a_r = a_r - a_{r-1}$ for $r \geq 1$ and $\nabla a_0 = 0$.
- **VIII.** **$S^i a$ and $S^{-i} a$ Numeric Functions:** If we denote the numeric functions $S^i a$ and $S^{-i} a$ by b and c respectively, then these may be defined as follows:

$$b_r = \begin{cases} 0, 0 \le r \le i-1 \\ a_{r-i}, r \ge i \end{cases}$$

and $c_r = a_{r+i}$ for $r \ge 0$. Here, i is some positive integer.

1.2.4 Some Important Examples

Example 1: If $a_r = \begin{cases} 0, 0 \le r \le 2 \\ 2^{-r} + 5, r \ge 3 \end{cases}$ and $b_r = \begin{cases} 3 - 2^r, 0 \le r \le 1 \\ r + 2, r \ge 2 \end{cases}$

(i) Find c_r if $c_r = a_r + b_r$. (ii) Find d_r if $d_r = a_r b_r$.

Sol. By the definition of sum and product of two given numeric functions, we may express c_r and d_r as follows:

(i) $c_r = a_r + b_r = \begin{cases} 3 - 2^r, 0 \le r \le 1 \\ 4, r = 2 \\ 2^{-r} + r + 7, r \ge 3 \end{cases}$ and (ii) $d_r = a_r b_r = \begin{cases} 0, 0 \le r \le 2 \\ r.2^{-r} + 2^{-r+1} + 5r + 10, r \ge 3 \end{cases}$

Example 2: Evaluate $a * b$ for the following numeric functions:

$a_r = \begin{cases} 1, 0 \le r \le 2 \\ 0, r \ge 3 \end{cases}$ and $b_r = \begin{cases} r + 1, 0 \le r \le 2 \\ 0, r \ge 3 \end{cases}$

Sol. For the given a_r and b_r, the numeric functions a and b are given by

$a = \{1,1,1,0,0,0,......\}$ and $b = \{1,2,3,0,0,0,.......\}$.

The convolution of a and b is a numeric function $c = a * b$ such that

$$c_r = a_o b_r + a_1 b_{r-1} + ... + a_r b_o = \sum_{i=0}^{r} a_i b_{r-i}$$

So, we have

$$c_0 = a_0 b_0 = (1)(1) = 1$$
$$c_1 = a_0 b_1 + a_1 b_0 = (1)(2) + (1)(1) = 3$$
$$c_3 = a_0 b_2 + a_1 b_1 + a_2 b_0 = (1)(3) + (1)(2) + (1)(1) = 6$$

Similarly, $c_3 = 5, c_4 = 3$ and $c_r = 0$ for $r \ge 5$.

$\therefore$ Numeric function c is given by $c = \{1,3,6,5,3,0,0,0,..........\}$,

where $c_r = \begin{cases} 1, r = 0 \\ 3, r = 1 \\ 6, r = 2 \\ 5, r = 3 \\ 3, r = 4 \\ 0, r \ge 5 \end{cases}$

Example 3: If the numeric function a is defined as $a_r = \begin{cases} 2, 0 \le r \le 3 \\ 2^{-r} + 5, r \ge 4 \end{cases}$. Then, evaluate

(i) $S^2 a$ (ii) $S^{-2} a$ (iii) Δa (iv) ∇a.

Sol. (i) For the numeric function a, the numeric functions $S^i a$ is given by

$$S^i a_r = \begin{cases} 0, 0 \le r \le i-1 \\ a_{r-i}, r \ge i \end{cases}$$

For $i = 2$, $S^2 a_r = \begin{cases} 0, 0 \le r \le 1 \\ a_{r-2}, r \ge 2 \end{cases}$

For given a_r, we have $S^2 a_r = \begin{cases} 0, 0 \le r \le 1 \\ 2, 2 \le r \le 5 \\ 2^{-(r-2)} + 5, r \ge 6 \end{cases}$

(ii) The numeric functions $S^{-i} a$ is defined as $S^{-i} a_r = a_{r+i}$ for $r \ge 0$.

$\therefore$ $S^{-2} a_r = a_{r+2}$ for $r \ge 0$.

For given a_r, we have $S^{-2} a_r = \begin{cases} 2, 0 \le r \le 1 \\ 2^{-(r+2)} + 5, r \ge 2 \end{cases}$

(iii) The numeric function Δa is defined as

$\Delta a_r = a_{r+1} - a_r$ for $r \ge 0$.

For given a_r, we have $\Delta a_r = \begin{cases} 0, 0 \le r \le 2 \\ 2^{-4} + 3, r = 3 \\ 2^{-(r+1)} - 2^{-r}, r \ge 4 \end{cases}$

(iii) The numeric function ∇a is defined as

$\nabla a_r = a_r - a_{r-1}$ for $r \ge 1$ and $\nabla a_0 = 0$.

For given a_r, we have $\nabla a_r = \begin{cases} 2, r = 0 \\ 0, 1 \le r \le 3 \\ 2^{-4} + 5, r = 4 \\ 2^{-r} - 2^{-(r-1)}, r \ge 5 \end{cases}$

1.2.5 Generating Function

Let S be a sequence with terms $S_0, S_1, S_2, ...$ .so on. Then, we may define the generating function $G(S, z)$ of the sequence S by the following infinite series:

$$G(S, z) = \sum_{n=0}^{\infty} S_n z^n = S_0 + S_1 z + S_2 z^2 + S_3 z^3 + ...\infty .$$

For example : Let the sequence S is $1^2, 2^2, 3^2, ...$ so on.

Then, $G(S, z) = 1^2.z^0 + 2^2.z^1 + 3^2.z^2 + ...\infty = \sum_{n=0}^{\infty} (n+1)^2 z^n$

1.2.6 Generating Functions of Some Standard Sequences

I. $S_n = a\,,\ n \geq 0$

$$G(S,z) = \sum_{n=0}^{\infty} S_n z^n = \sum_{n=0}^{\infty} a z^n = a\sum_{n=0}^{\infty} z^n$$

$$\Rightarrow\ G(S,z) = a(1 + z + z^2 + z^3 + \ldots\ldots\ldots\ldots\infty)$$

$$\Rightarrow\ G(S,z) = \frac{a}{1-z} \qquad [\because S_\infty = \frac{a}{1-r}]$$

II. $S_n = b^n\,,\ n \geq 0$

$$G(S,z) = \sum_{n=0}^{\infty} S_n z^n = \sum_{n=0}^{\infty} b^n z^n = \sum_{n=0}^{\infty} (bz)^n$$

$$\Rightarrow\ G(S,z) = 1 + bz + (bz)^2 + (bz)^3 + \ldots\ldots\ldots\ldots\infty)$$

$$\Rightarrow\ G(S,z) = \frac{1}{1-bz}$$

III. $S_n = cb^n\,,\ n \geq 0$

On the similar lines as above, it can be proved that

$$G(S,z) = \frac{c}{1-bz}$$

IV. $S_n = n$

$$G(S,z) = \sum_{n=0}^{\infty} S_n z^n = \sum_{n=0}^{\infty} n z^n = 0 + z + 2z^2 + 3z^3 + \ldots\ldots\ldots\ldots\infty$$

$$\Rightarrow\ G(S,z) = z(1 + 2z + 3z^2 + \ldots\ldots\ldots\ldots\infty)$$

$$\Rightarrow\ G(S,z) = \frac{z}{(1-z)^2}$$

Result: Generating function of sum of two sequences is equal to the sum of their generating functions. or

If $S_n = a_n + b_n$, then $G(S,z) = G(a,z) + G(b,z)$.

Proof: We can prove this result very easily.

As we know that $G(S,z) = \sum_{n=0}^{\infty} S_n z^n$

which gives $G(S,z) = \sum_{n=0}^{\infty} (a_n + b_n) z^n = \sum_{n=0}^{\infty} a_n z^n + \sum_{n=0}^{\infty} b_n z^n$

Therefore, $G(S,z) = G(a,z) + G(b,z)$

1.2.7 Operations on Sequences

If s,t are two sequences of natural number n, then we may define the following operations:

1. $(s+t)(n) = s(n) + t(n)$

   $cs(n) = c(s(n))$, where c is constant.

   $st(n) = s(n)t(n)$

2. **Convolution Operation:**

   $$(s * t)(n) = \sum_{r=0}^{n} s(r)t(n-r)$$

3. **Pop Operation** $s\uparrow$ (read as s pop):

   $(s\uparrow)n = s(n+1)$

4. **Push Operation** $s\downarrow$ (read as s push):

   $$(s\downarrow)n = \begin{cases} s(n-1), n > 0 \\ 0, n = 0 \end{cases}$$

Def : If s is a sequence of numbers, we define

$s\uparrow n = (s\uparrow(n-1))\uparrow$ if $n > 1$ and $s\uparrow 1 = s\uparrow$

Also, $s\downarrow n = (s\downarrow(n-1))\downarrow$ and $s\downarrow 1 = s\downarrow$.

In general, $(s\uparrow n)m = s(n+m)$

and $(s\downarrow n)m = \begin{cases} 0, m < n \\ s(m-n), m \ge n \end{cases}$

On the basis of above operations on sequences, we may state the following **Important results**:

1. $G(s+t,z) = G(s,z) + G(t,z)$
2. $G(cs,z) = cG(s,z)$, where c is constant.
3. $G(s*t,z) = G(s,z)G(t,z)$
4. $G(s\uparrow,z) = \dfrac{G(s,z) - s(0)}{z}$
5. $G(s\downarrow,z) = zG(s,z)$
6. $G(s\uparrow n,z) = \dfrac{G(s,z) - \sum_{r=0}^{n-1} s(r)z^r}{z^n}$
7. $G(s\downarrow n,z) = z^n G(s,z)$

1.2.8 Some Important Examples

Example 11.4: Write the generating function of the sequence $s_n = 3.4^n + 2.(-1)^n + 7$.

Sol. For the given sequence s_n,

$$G(s,z) = 3\left(\frac{1}{1-4z}\right) + 2\left(\frac{1}{1-(-1)z}\right) + 7.\frac{1}{1-z} = \frac{3}{1-4z} + \frac{2}{1+z} + \frac{7}{1-z}.$$

Example 2.5: Find the sequence whose generating function is $\frac{1}{1-z-z^2}$.

Sol. Here, $G(s,z) = \frac{1}{1-z-z^2}$ (1)

The roots of the equation $1-z-z^2 = 0$ are given by

$1-z-z^2 = 0 \Rightarrow z^2 + z - 1 = 0$

which gives $z = \frac{\sqrt{5}-1}{2}, \frac{-(\sqrt{5}+1)}{2}$

Let $\alpha = \frac{\sqrt{5}-1}{2}, \beta = \frac{-(\sqrt{5}+1)}{2}$ (2)

$\therefore z^2 + z - 1 = (z-\alpha)(z-\beta)$ or $1-z-z^2 = -(z-\alpha)(z-\beta)$

$$\Rightarrow \frac{1}{1-z-z^2} = \frac{-1}{(z-\alpha)(z-\beta)} = \frac{-1}{(z-\alpha)(\alpha-\beta)} + \frac{-1}{(\beta-\alpha)(z-\beta)}$$

$$\Rightarrow \frac{1}{1-z-z^2} = \frac{1}{\alpha-\beta}\left[\frac{1}{z-\beta} - \frac{1}{z-\alpha}\right] = \frac{1}{\alpha-\beta}\left[\frac{1}{\alpha-z} - \frac{1}{\beta-z}\right]$$

$$\therefore G(s,z) = \frac{1}{\alpha-\beta}\left[\frac{1}{\alpha\left(1-\frac{z}{\alpha}\right)} - \frac{1}{\beta\left(1-\frac{z}{\beta}\right)}\right]$$

which gives $s_n = \frac{1}{\alpha-\beta}\left[\frac{1}{\alpha}\left(\frac{1}{\alpha}\right)^n - \frac{1}{\beta}\left(\frac{1}{\beta}\right)^n\right] = \frac{1}{\alpha-\beta}\left[\left(\frac{1}{\alpha}\right)^{n+1} - \left(\frac{1}{\beta}\right)^{n+1}\right]$

Using the values of α and β, from (2), we get the required solution as

$$s_n = \frac{1}{\sqrt{5}}\left[\left(\frac{2}{\sqrt{5}-1}\right)^{n+1} - \left(\frac{-2}{\sqrt{5}+1}\right)^{n+1}\right]$$

Example 6: If $s_n = n^2 + 1$ and $t_n = n + 4$. Find $(s * t)(n), (s \uparrow 3)(n)$ and $(t \downarrow 2)(n)$.

Sol. $(s*t)(n)=\sum_{r=0}^{n} s(r)t(n-r)$

which gives $(s*t)(n)=\sum_{r=0}^{n}(r^2+1)(n-r+4)=\sum_{r=0}^{n}(n+4)r^2-r^3-r+n+4$

or $(s*t)(n)=(n+4)\sum_{r=0}^{n}r^2-\sum_{r=0}^{n}r^3-\sum_{r=0}^{n}r+(n+4)\sum_{r=0}^{n}1$

or $(s*t)(n)=(n+4)\dfrac{n(n+1)(2n+1)}{6}-\dfrac{n^2(n+1)^2}{4}-\dfrac{n(n+1)}{2}+n(n+4)$

or $(s*t)(n)=\dfrac{1}{6}n(n+1)(n+4)(2n+1)-\dfrac{1}{4}n^2(n+1)^2-\dfrac{1}{2}n(n+1)+n(n+4)$

Now, $(s\uparrow 3)(n)=s(n+3)=(n+3)^2+1=n^2+6n+10$

and $(t\downarrow 2)(n)=t(n-2)=n-2+4=n+2$ for $n\geq 4$.

Example 7: If $a(n)=n, b(n)=n/2, c(n)=2^n$. Find $G(a\uparrow 2,z)$, $G(b*b,z)$ and $G(2c,z)$.

Sol. For $a(n)=n$, $(a\uparrow 2)(n)=n+2=s_1+s_2$ where $s_1=n, s_2=2$

$\therefore G(a\uparrow 2,z)=G(s_1,z)+G(s_2,z)=\dfrac{z}{(1-z)^2}+\dfrac{2}{1-z}$

Further, $b(n)=\dfrac{n}{2}\Rightarrow G(b,z)=\dfrac{1}{2}\dfrac{z}{(1-z)^2}$

$\therefore G(b*b,z)=G(b,z)G(b,z)=\dfrac{1}{4}\dfrac{z^2}{(1-z)^4}$

Now, $c(n)=2^n\Rightarrow G(2c,z)=2G(c,z)=\dfrac{2}{1-2z}$

1.2.9 Summary

In this lesson, we have studied in detail about the discrete numeric functions and learnt the various operations on these functions. These numeric functions will help us to understand the concept of recurrence relations which will be discussed in the next lesson. Further, we have also gained the knowledge about the generating functions of some standard sequences and their various operations. The concept is made more clear with the help of suitable examples.

1.2.10 Self Check Exercise

1. Determine $a*b$ for the following numeric functions:

   $a_r=\begin{cases}1, 0\leq r\leq 2\\ 0, r\geq 3\end{cases}$ and $b_r=\begin{cases}r+1, 0\leq r\leq 2\\ 0, r\geq 3\end{cases}$

2. Write the sequence whose generating function is

(i) $\frac{3-5z}{1-2z-3z^2}$ (ii) $\frac{2}{1+z}+\frac{z}{(1-z)^2}$

3. If $S_n = 2^n, T_n = 3^n$. Then, find the convolution $S*T$ and verify that $G(S*T,z) = G(S,z)G(T,z)$

1.2.11 Suggested Readings

1. Norman L. Biggs, *Discrete Mathematics*, Oxford University Press.
2. Harmohan Sharma, Ganesh Kumar Sethi, *Discrete Mathematics,* Sharma Publications, Jalandhar.
3. C.L. Liu, Elements of Discrete Mathematics (Second Edition), McGraw Hill, International Edition, Computer Science Series, 1986.

LESSON NO. 1.3 **Author: Dr. Chanchal**

RECURRENCE RELATIONS

Structure:

3.0 Objectives

The prime goal of this lesson is to enlighten the basic concepts of recurrence relations and their solutions. During the study in this lesson, our main objectives are

- To study the basic concept of recursion.
- To understand the procedure of solving homogeneous and non-homogeneous recurrence relations.
- To understand the procedure of finding the generating function and sequence of a recurrence relation.

3.1 Introduction

The technique of defining a function, a set or an algorithm in terms of itself is known as **recursion**. An example is presented recursively, if every object is described in terms of two forms out of which one form is the **basis** for recursion which is written by a simple definition. The second form is written by a recursive description in which objects are described in terms of themselves i.e. the objects should be described in terms of simpler objects, where simpler means closer to the basis of recursion.

3.2 Some Recursive Definitions

I. The recursive definition of $\angle n$ is given by
$\angle 0 = 1$ and $\angle n = n(n-1)$.

Here n^{th} term is expressed as a function of previous term and $\angle 0 = 1$ is called basis.

II. The recursive definition of binomial coefficient $C(n,k)$ for $n \geq 0, k \geq 0, n \geq k$, is given by
$C(n,n) = 1, C(n,0) = 1$ and $C(n,k) = C(n-1,k) + C(n-1,k-1)$ if $n > k > 0$.

Here n^{th} term is expressed as a function of previous terms and $C(n,n) = 1, C(n,0) = 1$ are basis.

III. The recursive definition of a polynomial expression may be elaborated as:
Let S be the set of coefficients, then
(i) A zeroth degree polynomial is an element of S.
(ii) For $n \geq 1$, n^{th} degree polynomial expression is of the form $p(x)x + a$, where $p(x)$ is $(n-1)^{th}$ degree polynomial expression and $a \in S$.

IV. The recursive definition of **Fibonacci Sequence**, F is given by $F_0 = 1, F_1 = 1$ and $F_k = F_{k-2} + F_{k-1}$ for $k \geq 2$. Here, basis is the specification of first two numbers F_0 and F_1.

V. The recursive definition of positive integers can be given by the **Peano's Axioms**, as explained below:
Axiom 1: $1 \in \mathrm{N}$ i.e. 1 is a natural number.
Axiom 2: For each $n \in \mathrm{N}$, there exists a unique natural number n^*, called the successor of n given by $n^* = n+1$.
Axiom 3: 1 is not the successor of any natural number.
Axiom 4: If $m, n \in \mathrm{N}$ and $m^* = n^*$, then $m = n$.
Axiom 5: If $A \subset \mathrm{N}$, such that (i) $1 \in A$ and (ii) $n \in A \Rightarrow n^* \in A$, then $A = \mathrm{N}$. This axiom is also called the **Principle of Mathematical Induction**.
In the above definition, number 1 is the basis element and recursion is that if n is a positive integer, then its successor is also a positive integer.

3.3 Some Important Examples

Example 3.1: Determine $C(3,2)$ by the recursive definition of binomial coefficient.

Sol. By recursive definition: $C(n,n) = 1, C(n,0) = 1$ and
$C(n,k) = C(n-1,k) + C(n-1,k-1)$ if $n > k > 0$. (1)

Put $n = 3, k = 2$ in (1) and we get

$C(3,2) = C(2,2) + C(2,1)$ (2)

Now put $n = 2, k = 1$ in (1) and we get

$C(2,1) = C(1,1) + C(1,0) = 1 + 1 = 2$

So, from (2), $C(3,2) = 1 + 2 = 3$.

Example 3.2: Write $p(x) = 4n^3 + 2n^2 - 8n + 9$ in telescoping form.

Sol. Here, $p(x) = 4n^3 + 2n^2 - 8n + 9 = (4n^2 + 2n - 8)n + 9 = ((4n + 2)n - 8)n + 9$

$\Rightarrow p(x) = ((((4)n) + 2)n - 8)n + 9$ is the required telescoping form.

Example 3.3: If $B(0) = 2$ and $B(k) = B(k-1) + 3$ for $k \geq 1$. Evaluate $B(2)$ by the recursion formula and by the method of iteration.

Sol. By recursion formula:

$B(2) = B(1) + 3 = (B(0) + 3) + 3 = (2 + 3) + 3 = 5 + 3 = 8$

By iteration method:

$B(1) = B(0) + 3 = 2 + 3 = 5$

$B(2) = B(1) + 3 = 5 + 3 = 8$.

3.4 Recurrence Relation

For a numeric function $(a_0, a_1, a_2, ..., a_r, ...)$, an equation relating a_r, for any r, to one or more of the a_i's, $i < r$, is called a recurrence relation. It is also known as a **difference equation**. It is clear from the above definition that a step-by-step computation can be carried out to determine a_r from $a_{r-1}, a_{r-2}, ...$, and a_{r+1} from a_r, $a_{r-1}, ...$, and so on. It must be clear that the value of function at one or more points, known as the **boundary conditions**, must be given so that the computation procedure can be initiated. So, we may state here that the numeric function is also known as the solution of recurrence relation as it can be described by a recurrence relation together with an appropriate set of boundary conditions.

3.5 Linear Recurrence Relation with Constant Coefficients

A recurrence relation of the form

$$c_0 a_r + c_1 a_{r-1} + c_2 a_{r-2} + ... + c_k a_{r-k} = f(r) \quad (1)$$

is known as a linear recurrence relation with constant coefficients. Here, c_i's are constants and the above recurrence relation is of k^{th} order provided that the coefficients c_0 and c_k are non-zero.

Note: (i) Order of a recurrence relation is the difference between highest and lowest subscript. For example, $a_r + 8a_{r-2} = r^2 + 4$ is second order linear recurrence relation with constant coefficients.

(ii) The recurrence relation can also be determined from solution. For example, consider the closed form expression $S(k) = 9.2^k, k \geq 0$.

For $k \geq 1$, $S(k) = 9.(2.2^{k-1}) = 2.(9.2^{k-1}) = 2S(k-1)$

$\therefore S(k) - 2S(k-1) = 0$ and $S(0) = 9$ defines linear recurrence relation.

3.5.1 Homogeneous and Non-Homogeneous Recurrence Relation

A recurrence relation of the form

$$S(k) + C_1 S(k-1) + C_2 S(k-2) + ... + C_n S(k-n) = f(k)$$

is known as a (i) linear non-homogeneous relation if $f(k)$ is a function of k or a constant, (ii) homogeneous relation if $f(k) = 0$. Here $C_1, C_2, ...C_n$ are constants.

3.5.2 Characteristic Equation and Characteristic Roots

For n^{th} order linear recurrence relation of the form

$$S(k) + C_1 S(k-1) + C_2 S(k-2) + ... + C_n S(k-n) = f(k),$$

the characteristic equation is given by

$$a^n + C_1 a^{n-1} + C_2 a^{n-2} + ... + C_{n-1} a + C_n = 0.$$

Further, roots of the above characteristic equation are known as the characteristic roots and these roots may be real or imaginary.

3.6 How to Find Solutions of Recurrence Relation

Let the recurrence relation is of the form

$$S(k) + C_1 S(k-1) + C_2 S(k-2) + ... + C_n S(k-n) = f(k) \quad (1)$$

Case I : For $f(k) = 0$, the above relation (1) is homogeneous relation for which the solution can be obtained as

Step I. Write down the characteristic equation given by

$$a^n + C_1 a^{n-1} + C_2 a^{n-2} + ... + C_{n-1} a + C_n = 0 \quad (2)$$

Step II. Solve (2) and let the roots be $a_1, a_2, ..., a_n$.

Step III. If all the roots are different, then the general solution is given by

$$S(k) = b_1 a_1^k + b_2 a_2^k + ...b_n a_n^k.$$

If two real roots a_1, a_2 are such that $a_1 = a_2$, then solution is given by

$S(k) = (b_1 + b_2 k) a_1^k + b_3 a_3^k + ...b_n a_n^k$ and so on.

Case II : If $f(k)$ is a function of k, the above relation (1) is non-homogeneous relation whose solution consists of two parts out of which one is homogeneous solution and the other is particular solution and the general solution is given by $S(k)=S^{(h)}(k)+S^{(p)}(k)$. For the homogeneous solution $S^{(h)}(k)$, we put $f(k)=0$ and the solution is obtained as described under case I. The method for finding the particular solution $S^{(p)}(k)$ is explained below:

3.6.1 How to Find Particular Solution

Case I. When $f(k)$ is a constant

Let the particular solution is given by $S(k)=d$, then (1) becomes

$$d+C_1d+C_2d+.....+C_nd=f(k) \text{ which gives } d=\frac{f(k)}{1+C_1+C_2+.......+C_n}.$$

If $1+C_1+C_2+.....+C_n=0$, then it is a case of failure and we try for particular solution $S(k)=kd$. If for this too, the case fails, then the particular solution is taken as $S(k)=k^2d$ and so on.

Case II. When $f(k)$ is a linear function i.e., $f(k)=p_0+p_1k$

Let the particular solution is given by $S(k)=d_0+d_1k$, then (1) becomes

$$(d_0+d_1k)+C_1[d_0+d_1(k-1)]+C_2[d_0+d_1(k-2)]+.....+C_n[d_0+d_1(k-n)]=p_0+p_1k$$

On equating the coefficients of terms containing k and that of constant terms in the above expression, the values of d_0 and d_1 may be evaluated and then the particular solution $S(k)=d_0+d_1k$ is known.

Note: If $f(k)$ is an m^{th} degree polynomial of the form $f(k)=p_0+p_1k+p_2k^2+.....+p_mk^m$,

then the particular solution is given by $S(k)=d_0+d_1k+d_2k^2+.........+d_mk^m$.

It must be noted that if the particular solution contains any term similar to that of homogeneous solution, then the particular solution is multiplied by k .

Case III. When $f(k)$ is an exponential function i.e., $f(k)=pa^k$

Let the particular solution is given by $S(k)=da^k$, then (1) becomes

$$da^k+C_1da^{k-1}+C_2da^{k-2}+.....+C_nda^{k-n}=pa^k .$$

From the above equation, the value of d can be determined and then the particular solution is known. It must be noted that if the homogeneous solution contains a term containing a^k , then the particular solution is multiplied by k and given by $S(k)=dka^k$. Further, if the homogeneous solution contains a term containing ka^k , then the particular solution is given by $S(k)=dk^2a^k$ and so on.

3.7 How to Find Generating Function and Sequence of a Recurrence Relation

Let the recurrence relation is of the form

$S(n)+C_1S(n-1)+C_2S(n-2)+..................+C_rS(n-r)=0$ for $n \geq r.$

Step I. Multiply both sides by z^n and sum up terms from $n=r$ to ∞, we get

$$\sum_{n=r}^{\infty} S(n)z^n + C_1\sum_{n=r}^{\infty} S(n-1)z^n + C_2\sum_{n=r}^{\infty} S(n-2)z^n + ...+C_r\sum_{n=r}^{\infty} S(n-r)z^n = 0.$$

Step II. If $G(S,z)=\sum_{n=0}^{\infty} S(n)z^n$ be the generating function, then write each term in terms of $G(S,z)$.

Step III. Solve the equation for $G(S,z)$ and then with the help of standard generating functions (as discussed in the previous lesson 11), the sequence $S(n)$ can be obtained.

3.8 Some Important Examples

Example 9.4: Solve $S(k)-10S(k-1)+9S(k-2)=0, S(0)=3, S(1)=11.$

Sol. Put $S(k)=a^k$ in $S(k)-10S(k-1)+9S(k-2)=0$ and we obtain

$a^k - 10a^{k-1} + 9a^{k-2} = 0 \Rightarrow a^{k-2}\left(a^2-10a+9\right)=0$

$\Rightarrow a^2 - 10a + 9 = 0 \Rightarrow (a-1)(a-9)=0$

which gives $a=1, a=9$

$\therefore S(k) = C_1.1^k + C_2.9^k = C_1 + C_2.9^k$ ...(1)

Put $k=0$ and $k=1$ in (1), we get

$S(0) = C_1 + C_2.9^0 \Rightarrow 3 = C_1 + C_2$ ...(2)

and $S(1) = C_1 + C_2.9^1 \Rightarrow 11 = C_1 + 9C_2$ ...(3)

Now, subtracting (2) from (3), we have $8 = 8C_2 \Rightarrow C_2 = 1$

Put $C_2 = 1$ in (2), we obtain $C_1 = 2$.

So, $S(k) = 2 + 9^k$ is the required solution.

Example 3.5: Solve the recurrence relation: $\sqrt{a_n} = \sqrt{a_{n-1}} + \sqrt{a_{n-2}}$

Sol. Let $\sqrt{a_n} = b_n \Rightarrow a_n = b_n^{\ 2}$

$\therefore$ the given equation becomes: $b_n = b_{n-1} + b_{n-2}$

Let $b_n = m^n$

$\therefore m^n = m^{n-1} + m^{n-2} \Rightarrow m^n - m^{n-1} - m^{n-2} = 0 \Rightarrow m^{n-2}\left(m^2 - m - 1\right) = 0$

$$\Rightarrow m^2 - m - 1 = 0 \Rightarrow m = \frac{-(-1) \pm \sqrt{(-1)^2 - 4.1.(-1)}}{2.1} = \frac{1 \pm \sqrt{5}}{2}$$

$$\Rightarrow m = \frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}$$

which gives $b_n = C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n$ and the required solution is given by

$$a_n = b_n^{\ 2} = \left[C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n\right]^2$$

Example 3.6: Solve the recurrence relation: $s_n - 4s_{n-1} + 4s_{n-2} = 3n + 2^n$ with $s_0 = s_1 = 1$.

Sol. The solution of given non-homogeneous recurrence relation will consist of two parts i.e., homogeneous solution and particular solution.

Homogeneous Solution: The associated homogeneous relation is

$$s_n - 4s_{n-1} + 4s_{n-2} = 0$$

Its characteristic equation is $a^n - 4a^{n-1} + 4a^{n-2} = 0 \Rightarrow a^{n-2}\left(a^2 - 4a + 4\right) = 0$

$$\Rightarrow a^2 - 4a + 4 = 0 \Rightarrow (a-2)^2 = 0 \Rightarrow a = 2,2$$

$\therefore$ the homogeneous solution is $s_n^{(h)} = \left(c_1 + nc_2\right)2^n$

Particular Solution: Here $f(n) = 3n + 2^n$

Since base 2 in 2^n is a characteristic root repeated twice, therefore the particular solution is given by $s_n = cn + d + qn^2 2^n$

Using this value of s_n in the given equation, we obtain

$$cn + d + qn^2 2^n - 4\left(c(n-1) + d + q(n-1)^2 2^{n-1}\right) + 4\left(c(n-2) + d + q(n-2)^2 2^{n-2}\right) = 3n + 2^n$$

$$\Rightarrow cn + d + qn^2 2^n - 4cn - 4d + 4c - 2q\left(n^2 - 2n + 1\right)2^n + 4cn + 4d - 8c + q\left(n^2 - 4n + 4\right)2^2 = 3n + 2^n$$

$$\Rightarrow cn + d - 4c + 2q.2^n = 3n + 2^n$$

Equating the coefficients of like terms, we get

$$c = 3, d - 4c = 0, 2q = 1 \ \Rightarrow c = 3, d = 12, q = \frac{1}{2}$$

So, $s_n^{(p)} = 3n + 12 + \frac{1}{2}n^2 2^n = 3n + 12 + n^2 2^{n-1}$ is the required particular solution.

$\therefore$ general solution is

$$s_n = s_n^{(h)} + s_n^{(p)} = (c_1 + c_2 n)2^n + 3n + 12 + n^2 2^{n-1} \qquad ...(1)$$

It is given that $s_0 = s_1 = 1$. So, put $n = 0$ and $n = 1$ in the general solution and we get

$$s_0 = (c_1 + 0)2^0 + 0 + 12 + 0 \Rightarrow 1 = c_1 + 12 \Rightarrow c_1 = -11$$

and $s_1 = (c_1 + c_2)2 + 3 + 12 + 1 \Rightarrow 1 = (-11 + c_2)2 + 16 = -22 + 2c_2 + 16 \Rightarrow c_2 = \frac{7}{2}$

Put in (1), $s_n = \left(-11 + \frac{7}{2}n\right)2^n + 3n + 12 + n^2 2^{n-1}$

Example 3.7: Find the particular solution of $s_r - 5s_{r-1} + 6s_{r-2} = 3r^2$

Sol. Let $s_r^{(p)} = a + br + cr^2$

Using this value of s_r in the given equation, we obtain

$$a + br + cr^2 - 5\left(a + b(r-1) + c(r-1)^2\right) + 6\left(a + b(r-2) + c(r-2)^2\right) = 3r^2$$

$$\Rightarrow a + br + cr^2 - 5a - 5br + 5b - 5cr^2 - 5c + 10cr + 6a + 6br - 12b + 6cr^2 + 24c - 24cr = 3r^2$$

$$\Rightarrow 2cr^2 + 2br - 14cr + 2a - 7b + 19c = 3r^2$$

Equating the coefficients of like terms, we get

r^2 : $\quad 2c = 3 \Rightarrow c = \frac{3}{2}$

r : $\quad 2b - 14c = 0 \Rightarrow 2b = 14 \times \frac{3}{2} = 21 \Rightarrow b = \frac{21}{2}$

constant : $\quad 2a - 7b + 19c = 0 \Rightarrow 2a - 7 \times \frac{21}{2} + 19 \times \frac{3}{2} = 0 \Rightarrow 2a = \frac{90}{2} \Rightarrow a = \frac{45}{2}$

$$\therefore a_r^{(p)} = \frac{45}{2} + \frac{21}{2}r + \frac{3}{2}r^2$$

Example 3.8: By finding the generating function of sequence $S(n)$, find the solution of recurrence relation $S(n+2) - 7S(n+1) + 12S(n) = 0$ for $n \geq 0$ with $S(0) = 2, S(1) = 5$

Sol. The given recurrence relation can also be written as

$$S(n) - 7S(n-1) + 12S(n-2) = 0$$

Multiplying both sides by z^n and summing up terms from $n = 2$ to ∞, we get

$$\sum_{n=2}^{\infty} S(n)z^n - 7\sum_{n=2}^{\infty} S(n-1)z^n + 12\sum_{n=2}^{\infty} S(n-2)z^n = 0$$

$$\Rightarrow G(S,z) - S(0) - S(1)z - 7z\sum_{n=2}^{\infty} S(n-1)z^{n-1} + 12\sum_{n=2}^{\infty} S(n-2)z^{n-2} = 0$$

$$\Rightarrow G(S,z)-2-5z-7z[G(S,z)-S(0)]+12z^2G(S,z)=0$$
$$\Rightarrow G(S,z)-2-5z-7z[G(S,z)-2]+12z^2G(S,z)=0$$
$$\Rightarrow G(S,z)-2-5z-7zG(S,z)+14z+12z^2G(S,z)=0$$
$$\Rightarrow (1-7z+12z^2)G(S,z)=2-9z$$
$$\Rightarrow G(S,z)=\frac{2-9z}{1-7z+12z^2}=\frac{2-9z}{(1-3z)(1-4z)}=\frac{2-3}{(1-3z)\left(1-\frac{4}{3}\right)}+\frac{2-\frac{9}{4}}{\left(1-\frac{3}{4}\right)(1-4z)}=\frac{3}{1-3z}-\frac{1}{1-4z}$$

which gives $S(n)=3.3^n-4^n=3^{n+1}-4^n$

3.9 Summary

In this lesson, we have tried to elaborate the concept of recursion with the help of recursive definitions. On the same ground, we have learnt about the recurrence relations and generating functions of recurrence relations. Further, this lesson teaches us the procedure to find out the general and particular solutions of recurrence relations. During the study, it is found that the numeric function is also known as the solution of recurrence relation. The concept is made more clear with the help of suitable examples.

3.10 Self Check Exercise

1) Write short note on recursion.
2) Solve $s_n+5s_{n-1}=9, s_0=6$.
3) Solve $s_n-4s_{n-1}+4s_{n-2}=(n+1)2^n$.
4) If the solution of recurrence relation $as_n+bs_{n-1}+cs_{n-2}=6$ is 3^n+4^n+2, then find a,b,c.
5) Define the Febonacci sequence and find its generating function.
6) By finding the generating function of sequence $S(n)$, find the solution of recurrence relation $S(n)+3S(n-1)-4S(n-2)=0$ for $n\geq 2$ with $S(0)=3, S(1)=-2$.

3.11 Suggested Readings

1. Norman L. Biggs, *Discrete Mathematics*, Oxford University Press.
2. Harmohan Sharma, Ganesh Kumar Sethi, *Discrete Mathematics,* Sharma Publications, Jalandhar.
3. C.L. Liu, Elements of Discrete Mathematics (Second Edition), McGraw Hill, International Edition, Computer Science Series, 1986.