NEERAJ

ADVANCED DISCRETE MATHEMATICS

By:

Subhash G. Deo, B.E. (Elect.), B.A. (Hon. Maths), P.G.D.

Reference Book

Including

Solved Question Papers

New Edition

NEERAJ PUBLICATIONS

(Publishers of Educational Books)

(An ISO 9001 : 2008 Certified Company)

1507, Ist Floor, NAI SARAK, DELHI - 110006

Ph.: 011-23260329, 45704411, 23244362, 23285501

E-mail: **info@neerajignoubooks.com**

Website: **www.neerajignoubooks.com**

Price ₹ 160/-

Published by:

NEERAJ PUBLICATIONS

Admn. Office : **Delhi-110 007**
Sales Office : **1507, 1st Floor, Nai Sarak, Delhi-110 006**
E-mail: **info@neerajignoubooks.com** *Website*: **www.neerajignoubooks.com**

Reprint Edition	Typesetting by: ***Competent Computers***	Printed at: ***Novelty Printer***

Notes:

1. *For the best & upto-date study & results, please prefer the recommended textbooks/study material only.*
2. *This book is just a Guide Book/Reference Book published by NEERAJ PUBLICATIONS based on the suggested syllabus by a particular Board /University.*
3. *The information and data etc., given in this Book are from the best of the data arranged by the Author, but for the complete and upto-date information and data etc., see the Govt. of India Publications/textbooks recommended by the Board/University.*
4. *Publisher is not responsible for any omission or error though every care has been taken while preparing, printing, composing and proof reading of the Book. As all the Composing, Printing, Publishing and Proof Reading etc. are done by Human only and chances of Human Error could not be denied. If any reader is not satisfied, then he is requested not to buy this book.*
5. *In case of any dispute whatsoever the maximum anybody can claim against NEERAJ PUBLICATIONS is just for the price of the Book.*
6. *If anyone finds any mistake or error in this Book, he is requested to inform the Publisher, so that the same could be rectified and he would be provided the rectified Book free of cost.*
7. *The number of questions in NEERAJ study materials are indicative of general scope and design of the question paper.*
8. *Question Paper and their answers given in this Book provide you just the approximate pattern of the actual paper and is prepared based on the memory only. However, the actual Question Paper might somewhat vary in its contents, distribution of marks and their level of difficulty.*
9. *Any type of ONLINE Sale/Resale of "NEERAJ BOOKS/NEERAJ IGNOU BOOKS" published by "NEERAJ PUBLICATIONS" on Websites, Web Portals, Online Shopping Sites, like Amazon, Flipkart, Ebay, Snapdeal, etc. is strictly not permitted without prior written permission from NEERAJ PUBLICATIONS. Any such online sale activity by an Individual, Company, Dealer, Bookseller, Book Trader or Distributor will be termed as ILLEGAL SALE of NEERAJ IGNOU BOOKS/NEERAJ BOOKS and will invite legal action against the offenders.*
10. *Subject to Delhi Jurisdiction only.*

CONTENTS

ADVANCED DISCRETE MATHEMATICS

■ ■

Sample Preview
of the
Solved
Sample Question
Papers

Published by:

NEERAJ
PUBLICATIONS
www.neerajbooks.com

QUESTION PAPER

(June – 2017)

(Solved)

ADVANCED DISCRETE MATHEMATICS

Time: 2 Hours] *[Maximum Marks: 50*

Note: *Question no.* **1** *is* **compulsory.** *Attempt any* **three** *questions from the rest.*

Q. 1. ***(a)*** **Find the order and degree of the following recurrence relation. Also determine whether they are homogeneous or non-homogeneous.**

(i) $a_n = a_{n-1} + a_{n-2} + ... + a_0$

Ans. $a_n = a_{n-1} + a_{n-2} + ... + a_0$

is homogeneous, has no order, but has degree1.

(ii) $a_n = na_{n-1} + (-1)^n$

Ans. $a_n = na_{n-1} + (-1)^n$

is non-homogeneous, has order 1 and degree 1.

(iii) $a_n = a_{n-1} + a_{n-2}$

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

is homogeneous, has order 2 and degree 1.

(b) **Solve the following recursion relation using characteristic equation:**

$$t_n = 4t_{n-1} - 3t_{n-2} \text{ for } n > 1$$
$$t_0 = 0$$
$$t_1 = 1$$

Ans. $t_n = 4t_{n-1} - 3t_{n-2}$ for $n > 1$

$t_0 = 0$

$t_1 = 1$

The characteristic equation of the recurrence relation is $t^2 - At - B = 0$

where $A = 4$

$B = -3$ [By the given relation]

$t^2 - 4t + 3 = 0$

$t^2 - 3t - t + 3 = 0$

$t(t-3) - 1(t-3) = 0$

$(t-1)(t-3) = 0$

$t = 1$ and $t = 3$

$\therefore$ therefore the roots are distinct roots & real

$t_1 = 1$ and $t_2 = 3$

So the solution is

$tn = at_1n + bt_2n$

$tn = a_1n + b_3n$

Therefore, for to $\Rightarrow$

$t_0 = a\,1.0 + b\,3.0$

$0 = a + b$ *(i)*

Similarly for $t_1 \Rightarrow$

$t_1 = a\,1.1 + b\,3.1$

$1 = a + 3b$ *(ii)*

Solving these two equations *(i)* & *(ii)* we have

$a = \frac{-1}{2}$ and $b = \frac{1}{2}$

Putting these values in the solution we have

$$\boxed{tn = \frac{-1}{2}t_1n + \frac{1}{2}t_2n}$$

This is the final solution.

(c) **State and prove the handshaking theorem.**

Ans. Handshaking Theorem

If G is a (p, q) – graph with

$V(G) = \{v_i, ..., v_p\}$ and if $d_i = dG(v_i)$,

$1 \le i \le p$, the $2q = \sum_{i=1}^{p} d_i$.

That is, the sum of the degrees of the vertices of G is twice the no of edges.

Proof. Consider the set S = $\{x, e)$: $n \in v$ (G), $e \in E(G)$,

x is an end point of e$\}$.

Now we can choose a vertex $0_i \in V$ in p ways. Since $d_i = d(v_i)$, there are exactly d_i edges incident with the vertex v_i. These edges indicate d_i elements of the set S. Adding all the verties of G, we obtain

$$|S| = \sum_{i=1}^{p} d_i.$$

Next we can choose an edge e in E(G) in q ways.

This edge has exatly two endpoints & they indicate two elements of S. Adding every edge $e \in E(G)$, we obtain $|S| = 2q$

This is because every edge is counted twice, once for each of two vertices it concists of. Equation *(i)* & *(ii)*, we have the desired result.

***(d)* Define the following symbols:**

***(i)* δ (G)**

Ans.*(d) (i)* δ (G)

If G = (V.E) is a (*p. q*) – graph, then

δ (G) – min $\{d_G(x) : x \in V(G)\}$ is called the minimum vertex degree of G..

δ (G) is non-negative integer.

***(ii)* <S > G**

Ans. *(ii)* <S > G

Let G be a graph and Let S $\subseteq$ V (G) By the subgraph of the graph G, induced by the set S. We mean the subgraph H with V(H) = S and the edge set consisting of those edges of G which are joining the vertices in S. That is, E(H) $\{xy: x \neq y, x \in S\, y \in S, xy \in E(G)\}$. We denoted H by < S> G

***(iii)* W_n**

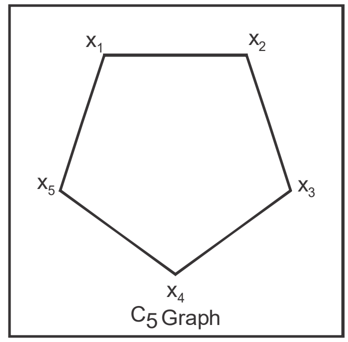
Ans. W_n A walk in a graph G is a finite sequence $W_n = \{v_0, e_1, v_1, e_k, ...e_k <, v_1 <, e_n\, v_n\}$ here $v_0, v_1 . . . v_k$ v_n are vertices of G and $e_1, e_2 . . . e_k, e_n$ are edges joining the vertices.

Q. 2. *(a)* What is meant by complement of a graph? Find the complement of the C_5 graph (*i.e.* $\overline{C}_5$).

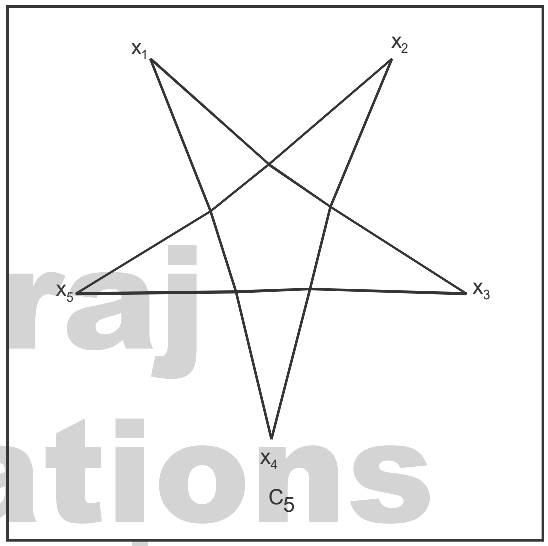
Sol. Complement of a graph

Let G = (V.E) be a (*p. q*) graph. Its complement $\overline{G}$, is the graph with V($\overline{G}$) = V(G) and E ($\overline{G}$) = $\{xy : xy \notin E(G), x, y, EV(G)\}$.

If is to be noted that $\overline{G}$ is a (*p. q*) – graph, where $\overline{q}$ = (no. of pairs of elements of V) – *q*. In a set V with *p* elements, there can be C (*p*, 2) = $\frac{p(p-1)}{2}$ such pairs of elements, hence $\overline{q} = \frac{p(p-1)}{2} - q$.



C5 Graph



C_5

***(b)* What is a complete graph?**

Ans. Complete graph–A complete graph is a graph in which any two vertices are adjacent, *i.e.*, each vertex is joined to every other vertex by an edge. A complete graph on *n* vertices is denoted by K_n.

Ex. K_1, K_2 has two vertices & an edge

$K_1 \int K_2$

***(c)* Find the generating function for the sequence $0^2, 1^2, 2^2, 3^2, ...$**

Sol. $0^2, 1^2, 2^2, 3^2, ...$

Begin with the generating function for $\{1, 1, 1, 1, ...\}$, differentiate, multi, *xy* by *x* and then differentiate and multiply by x once more $\{1, 1, 1, 1, ...\} \longleftrightarrow \frac{1}{1-x}$

$$\{1, 2, 3, 4, ...\} \longleftrightarrow \frac{d}{dx}\frac{1}{1-x} = \frac{1}{(1-x)^2}$$

Sample Preview
of
The Chapter

Published by:

NEERAJ
PUBLICATIONS
www.neerajbooks.com

ADVANCED DISCRETE MATHEMATICS

RECURRENCES

Recurrence Relations

1

INTRODUCTION

Earlier, we have studied to solve different types of combinatorial problems using various tools. However, there are still many other problems involving counting which cannot be solved only with the techniques we have learnt till now. For example, consider the problem of counting number of binary strings of length n that do not have two consecutive ones (or zeros).

Let us denote the number of such binary strings as a_n, then $a_1 = 2$ and $a_2 = 3$. Also, it can be shown that $a_n = a_{n-1} + a_{n-2}$ for $n \geq 2$. This is an example of a recurrence relation. Here it relates the values of a_n, a_{n-1} and a_{n-2}. We can, in fact, find an explicit expression for a_n using this relation. In this chapter, we will study how to formulate such recurrence relations for solving combinational problems. Introduced here are recurrence relations of three well-known examples, the Fibonacci Recurrence, Towers of Kashi (or Hanoi) and the number of ways of parenthesising an expression.

Later, we take up more examples on formulating the recurrences.

A formal definition of recurrence relation then follows alongwith terminology related to recurrences e.g. order and degree of a recurrence relation.

Finally, we analyse the Divide and Conquer techniques used in the design of algorithms that lead to recurrences in natural way. The focus is on recurrences associated with algorithms for locating the maximum and minimum elements of a list, quick multiplication of integers etc. Topics covered in this chapter are:

1. Definition of a recurrence relation;
2. Examples of recurrence relations;
3. Setting up of recurrence relations;
4. Writing recurrences for divide and conquer algorithms.

CHAPTER AT A GLANCE

THREE RECURRENT PROBLEMS

We start our study of recurrence relations with three interesting problems that provide a good idea of the subject-matter. The first two of these problems are famous and popular. All three have a solution based on the concept of recurrences. This is because the solution to each of them depends on the solution of smaller instances of the same problem.

Problem 1: Rabbits and the Fibonacci Numbers *n*: The problem of breeding rabbits was originally put forward by **Leonardo di Pisa,** better known as **Fibonacci.** He posed it in 1202 in his book **Liber abaci.** The problem is one pair of rabbits, one male and one female, live on an island. They start breeding at the end of two months and produce a pair of rabbits of opposite sex at the end of each month thereafter. Suppose f_n is the number of pairs of rabbits after n months. Then $f_1 = 1$. The rabbits start breeding only after two months and the young ones will be born one month afterwards. Hence, young ones appear only at the end of third month. Clearly, the number of pairs of rabbits is still 1 at the end of the second month i.e $f_2 = 1$. At the end of the third month, the pair would have given birth to one more pair. Table 1 gives further details. To know the number of pairs after n months, we have to add the number of pair after $n - 1$ months to the number of pairs born in the nth month. However, the newborns come from pairs at least two months old, i.e. from the pairs that are already there after $n - 2$ months; there are f_{n-2} of these. Hence, the sequence $\{f_n | n \geq 1\}$ satisfies the condition $f_n = f_{n-1} + f_{n-2}$ if $n \geq 3$. These f_n are called **Fibonacci numbers.**

Table 1: Number of Rabbits on the Island

Months	Reproducing Pairs (at least two months old)	Young Pairs (not more than two months old)
1		
2		
3		
4		
5		
6		

Obviously, we have not yet solved the problem. However, we have a uniquely defined sequence describing its future members in term of present members. We may also define f_n as a function of n. (See Q. 1)

We will take up the Fibonacci sequence again later, but in the meantime we consider another famous recurrence problem.

Problem 2: The Tower of Kashi (or Hanoi)

This problem was first stated by French mathematician Edouard Lucas in 1883. There is a tower of eight discs, initially stacked in decreasing size on one of three pegs as shown in Figure 1.

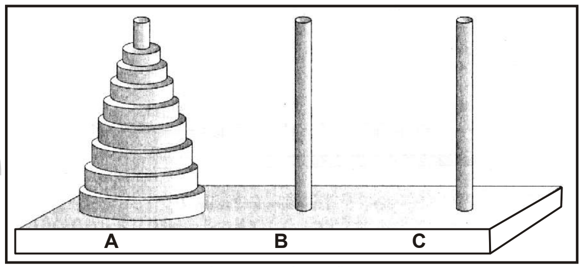
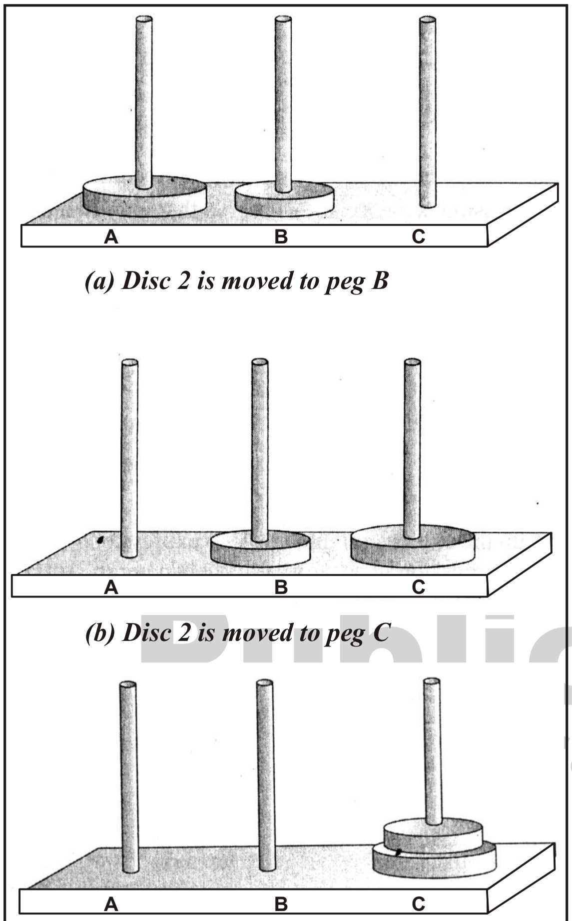


Fig. 1: Initial position of the towers of Kashi (Hanoi) problem

The objective is to transfer the complete tower to one of the other two pegs, moving only one disc at a time without at any time moving a larger disc onto a smaller one. Lucas embellished this tower with a legend about a much bigger **Tower of Brahma,** which has 64 discs of gold resting on three diamond needles. "At the beginning of time", he wrote , "God placed these golden diamond needles", "God placed these golden discs on the first needle and said that a group of priests should transfer them to a third, according to the rules above. The Tower will crumble and the world will come to an end once that task is finished."

We now generalise the problem and check the results for n discs. Let T_n be the minimum number of moves that will transfer n discs from one peg to another satisfying the given rules. Obviously, $T_1 = 1$, and $T_2 = 3$ (Figure 2). We can first experiment with two disks to find a general strategy: We first transfer the $n - 1$ smallest discs to B (using T_{n-1} moves), then move the largest (needing one move) to C. A is now empty and we can use it to transfer the discs on peg B to C. In this way, we can transfer n discs.



(a) Disc 2 is moved to peg B

(b) Disc 2 is moved to peg C

(c) Disc 2 is moved to peg C

Fig. 2: The moves involved in transferring 2 discs

(for $n \geq 2$) in at most $2T_{n-1} + 1$ moves. Hence, $T_n \leq 2\,T_{n-1} + 1$, if $n \leq 2$. We have used "$\leq$" instead of " = " here. This is because our strategies proves only that $2T_{n-1} + 1$ move are sufficient but it does not prove that we can transfer the discs with lesser number of moves. In fact, it is not possible to do so. At some point, we have to move the largest disc. At that time, the $n - 1$ smallest must be on a single peg because the largest disc is on one peg and it is now being moved to the third empty peg to be at its bottom-most position. It has taken at least T_{n-1} moves to put them on that peg. After moving the largest disc for the last time, we have to now transfer the $n - 1$ smaller discs (which are again on a single peg) back onto the largest disc. This also needs T_{n-1} moves. Therefore, $T_n \geq 2T_{n-1} + 1$ if $n \geq 2$. Both the inequalities, $T_n \geq 2\,T_{n-1} + 1$ and $T_n \leq 2\,T_{n-i} + 1$ can be true only if $T_n = 2T_{n-1} + 1$.

As a matter of fact, the priests of Varanasi will require a minimum of $2^{64} - 1 = 18446\ 744\ 073\ 709\ 551\ 615$ moves to transfer the golden discs. At the superfast rate of one move per second, they will need more than 5×10^{11} years to complete the task! The doomsday for us is quite distant!

Example 1: Suppose two candidates A and B get the same number of votes, n each, in an election. The counting of votes is mostly carried out in an arbitrary order and as such during the counting A may lead for sometime and B may lead for sometime. The number of ways to count the votes such that A does not lag behind B at any time is the nth Catalan number. We can represent a vote for A by + and a vote for B by –. With this the nth Catalan number is the number of sequences of pluses and minuses such that the number of pluses are equal to number of minuses at any stage of the sequence. We can call such a sequence an admissible sequence. We now consider a special case where 8 votes are polled, 4 for A and 4 for B. One voting sequence where A never trails is (+, –, +, +, –, –, +, –). If we omit last two terms, we have the sequence (+, –, +, +, –, –) with 3 pluses and 3 minuses i.e. there are at least as many pluses as there are minuses. If the last term is not considered, we obtain (+, –, +, +, –, –, +), where there are 4 pluses and 3 minuses, i.e. pluses are more than minuses.

Example 2. In computer science a data structure called **Stack** is often used. A stack is a list which can be changed by insertions or deletions at its top. An insertion is termed a **push** and a deletion is termed a **pop**. A sequence of pushes and and pops of length $2n$ is said to be admissible if there are n pushes and n pops and at each stage of the sequenc there occur at least as many pushes as pops. Let there be a string 123...*n* of the set N = {1, 2, 3, ..., *n*} and an admissible sequence of pushes and pops of length 2*n*. Each push in the sequence moves the last element in the input string to the stack and every pop moves the element on the top of the stack to the beginning of the output string. After doing *n* pushes and pops, the output string is permutation of N called a **stack permutation**. The number of stack permutations of 123... *n* is the *n*th Catalan number. Suppose, we represent a pop by *a* and *a* + push by *a* –. Clearly, every admissible sequence of pops and pushes relates to an admissible sequence of pluses and minuses. Let *n* = 4 and an admissible sequence of pops and pushes as (+, –, +, +, –, –, +, –). The stack permutation corresponding to this is shown in Table 2.

Table 2: Stack permutation corresponding to admissible sequence (+, –, +, +, –, –, +, +, –)

Sequence	Input String	Stack	Output String
+ (Push 4)	123	[4]	Empty
– (Pop 4)	123	[ ]	4
+ (Push 3)	12	[3]	4
+ (Push 2)	1	[23]	4
– (Pop 2)	1	[3]	24
– (Pop 3)	1	[ ]	324
+ (Push 1)	Empty	[1]	324
– (Pop 1)	Empty	[ ]	1324

Hence, the permutation derived is 1324. It is a stack permutation of size 4.

We find that in all the three examples discussed, we can express the *n*th term of a sequence in terms of one or more previous terms and a function of *n*. This provides us with a method to calculate the terms of the sequence accurately, though it may take sometime. When the relation between the terms is in a fair form, we can even obtain the recurrence. i.e. express the *n*th term as a function of *n*. We will soon learn how to solve the three recurrences and get their function.

MORE RECURRENCES

We have so far studied a variety of recurrent problems. It is now time to take a close look at setting up recurrence relations for combinational problems. When we trying to determine recurrence, we in fact, try to describe the counting inductively. In general, the recurrence relation leads to an alternate method of solution.

Example: Suppose C_n is the number of comparisons required to sort a list of *n* integers. We want to find a recurrence relation for C_n. We first locate the minimum of the *n* elements. This is going to be the first element of the list. We compare the first two elements and obtain the smaller among the two. We then compare it with the third element, and so on. To know the minimum of *n* elements we must make *n*–1 comparisons. As an example, if we want to know the minimum of the list 6, 5, 7, 4, 8 four comparisons would be needed as shown in Figure 3.

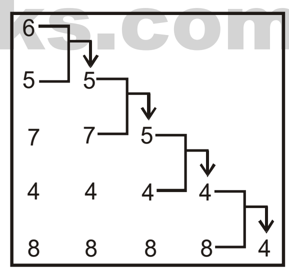


Fig. 3: Comparisons for finding the minimum of 6, 5, 7, 4, 8

We now put the minimum element found as the first element of the list. Thus we continue to sort the remaining *n* – 1 elements with C_{n-1} comparisons and append it after first element. Hence, *n* – 1 + C_{n-1} comparisons have to done to sort a list of *n* elements.

or $C_n = C_{n-1} + n - 1$