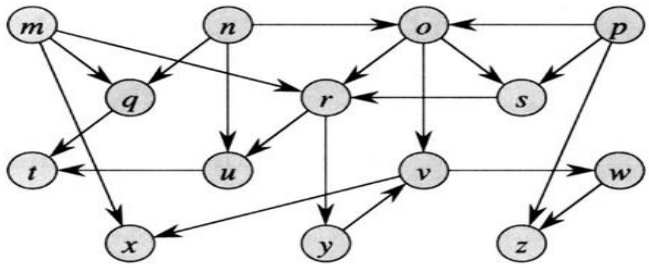
Program: BE

Date	July 2024	Maximum Marks	10+50
Course Code	**CD343AI**	Duration	120 min
Sem	IV	CIE-II(Scheme and solution)	

Design and Analysis of Algorithms

(Common to AIML/CSE/CD/CY/ISE)

Sl. No.	Questions	M
	PART A	
1.1	Yes, Run the DFS or BFS from each vertex to find the path to all other vertices.	**2**
1.2	27	**2**
1.3	O(n)	**2**
1.4	No matter how many edges are there in the graph the Floyd's Algorithm runs for $O(V^3)$ times therefore it is best suited for Dense graphs.	**2**
1.5	1. Limited to Integer Sorting 2. Memory Consumption 3. Range Limitation 4. Not Adaptive to the data's distribution or partially sorted input 5. Stability Concerns 6. Preprocessing Overhead	**2**
	PART B	
1a	Apply DFS traversal to find the topological order of the graph shown in figure 1a from the vertex p (break the ties by the alphabetical order of the vertices)  Topological order: n m q p o s r y v x w z u t	**06**
1b	**Brute force :** Time Complexity: $O(N^2)$----0.5M	**04**

Instance simplification method(Presorting)
Time Complexity: O(N * logN)----0.5M
Comparison with algorithm ---02M

2a

Array A[0---6]		94	73	26	11	05	77	31
Initially	Count[]	0	0	0	0	0	0	0
After pass	i=0	**6**	0	0	0	0	0	0
	i=1		**4**	0	0	0	1	0
	i=2			**2**	0	0	2	1
	i=3				**1**	0	3	2
	i=4					**0**	4	3
	i=5						**5**	**3**
	Index of the item in the final array	6	4	2	0	3	5	1
Array S[0----6]		05	11	26	31	73	77	94

06

2b

ALGORITHM *Warshall*$(A[1..n, 1..n])$
//Implements Warshall's algorithm for computing the transitive closure
//Input: The adjacency matrix A of a digraph with n vertices
//Output: The transitive closure of the digraph
$R^{(0)} \leftarrow A$
for $k \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
      $R^{(k)}[i, j] \leftarrow R^{(k-1)}[i, j]$ **or** $(R^{(k-1)}[i, k]$ **and** $R^{(k-1)}[k, j])$
return $R^{(n)}$

---------**02M**

Proof--------**02M**

04

3a

Heap construction **------01M**

8	12	15	3	5	1	43	-7
8	12	**43**	3	5	1	**15**	-7
43	12	**8**	3	5	1	15	-7
43	12	**15**	3	5	1	**8**	-7

Maximum deletion----**02M**
All Heap construction iteration in the process of sorting **------03M**

43	12	15	3	5	1	8	-7
-7	12	15	3	5	1	8	43
15	12	8	3	5	1	-7	
-7	12	8	3	5	1	15	
12	5	8	3	-7	1		
1	5	8	3	-7	12		
8	5	1	3	-7			
-7	5	1	3	8			
5	3	1	-7				
-7	3	1	5				
3	-7	1					

06

	1 -7 3 -7 1	
3b		**04**

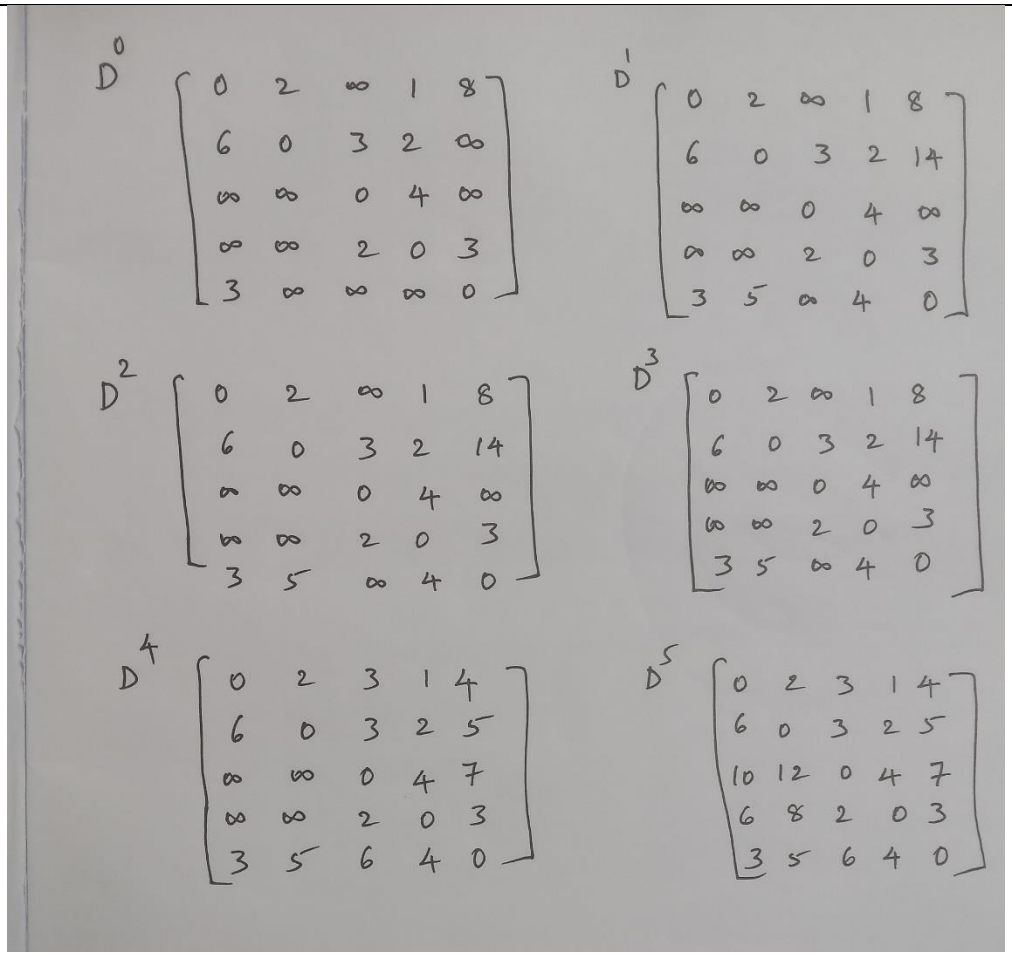
	0	1	2	3
0	1			
1	1	1		
2	1	2	1	
3	1	3	3	1
4	1	4	6	4
5	1	5	10	10

----------**02M**

The value of C(n, k) can be recursively calculated using the following standard formula for Binomial Coefficients. ----------**02M**

C(n, k) = C(n-1, k-1) + C(n-1, k)

C(n, 0) = C(n, n) = 1

4		**10**

Bad Symbol Table

K	N	A	R	*
4	1	3	2	11

Good Suffix Table

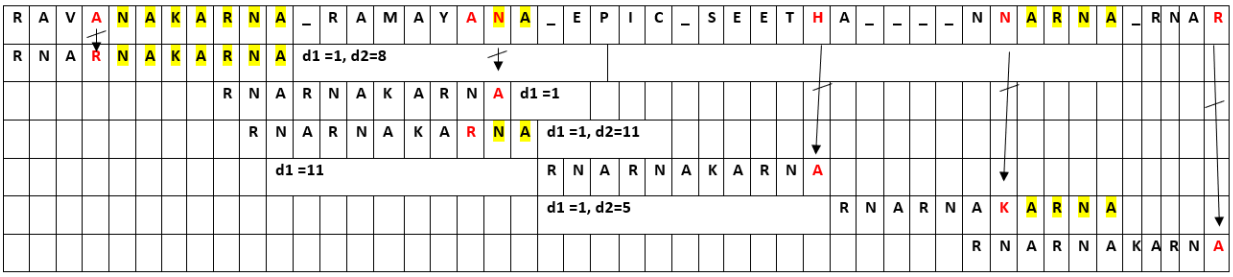
K	Pattern	d_2
1	RNARNAKARNA	3
2	RNARNAKARNA	11
3	RNARNAKARNA	8
4	RNARNAKARNA	5
5	RNARNAKARNA	8
6	RNARNAKARNA	8
7	RNARNAKARNA	8
8	RNARNAKARNA	8
9	RNARNAKARNA	8
10	RNARNAKARNA	8

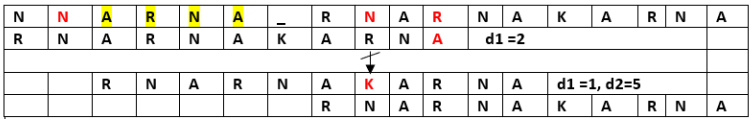
5

10

Bad symbol table -----01M

Good suffix table------04M





Above steps----------05M