

Course/ Batch: BTech/ SCSET
Course Code: CSET244
Semester: Even
Session: 2024-2025

Course Type: Core
Course Name: Design and Analysis of Algorithms

Tutorial Assignment: Week 11

Tutorial title: Dynamic Programming

Q1. Let A_1, A_2, A_3 , and A_4 , be four matrices of dimensions 10×5 , 5×20 , 20×10 , and 10×5 , respectively. Find the minimum number of scalar multiplications required to find the matrix product.

Solution: Use the equation $c_{i,j} = \min_k (c_{i,k} + c_{k,j} + p_{i-1}p_kp_j)$ for $k = i$ to $j - 1$ to find $c_{i,j}$ and the optimal decision index $d_{i,j}$. The final matrix will be

c	1	2	3	4
1	0	1000	1500	1500
2		0	1000	1250
3			0	1000
4				0

d	1	2	3	4
1	0	1	1	1
2		0	2	3
3			0	3
4				0

minimum number of scalar multiplications required $= c_{1,4} = 1500$. The multiplication order will be $A_1(A_2A_3)A_4$

Q2. Compute the binomial coefficient $C(6,3)$ using dynamic programming based on Pascal's Identity.

k->	0	1	2	3
n= 1	1	1		
n = 2	1	2	1	
n = 3	1	3	3	1
n = 4	1	4	6	4
n = 5	1	5	10	10
n = 6	1	6	15	20

$$C(6,3) = 20.$$

Q3. Consider the dynamic programming approach to calculate the binomial coefficient $C(n, k)$ using Pascal's Identity. Which of the following statements is TRUE?

- (A) The base case for the recursion is $C(n, k) = 1$ if $n = k$ or $k = 0$.
- (B) The recurrence relation used is $C(n, k) = C(n + 1, k - 1) + C(n + 1, k)$.
- (C) The time complexity of the dynamic programming solution to calculate $C(n, k)$ is exponential in terms of n .
- (D) The dynamic programming approach avoids redundant calculations by storing the results of subproblems.

Correct : A and D

Q4. If you implement the bottom-up dynamic programming approach to find $C(n, k)$, approximately how many addition operations will be performed to fill the DP table?

- (A) $O(n)$
- (B) $O(k)$
- (C) $O(n \times k)$
- (D) $O(n + k)$

Correct : C: The DP table typically has dimensions $(n+1) \times (k+1)$ (or $(n+1) \times (n+1)$ if k can go up to n). Each entry (except the base cases) involves one addition. Thus, the number of additions is proportional to the number of entries calculated, which is $O(n \times k)$.

Q5. Consider a directed weighted graph with 4 vertices labeled 1, 2, 3, and 4. The initial adjacency matrix representing the direct edge weights is given below (where ∞ represents no direct edge):

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

After running the Floyd-Warshall algorithm, what is the shortest path weight from vertex 4 to vertex 3?

Solution: Use the equation $d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}$ for $k \geq 1$ to find the matrix $D^{(k)}$

$D^{(0)}$	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

$D^{(1)}$	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

$D^{(2)}$	1	2	3	4
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

$D^{(3)}$	1	2	3	4
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

$D^{(4)}$	1	2	3	4
1	0	3	4	1
2	5	0	1	6
3	4	7	0	5
4	7	2	3	0

Shortest path weight from vertex 4 to vertex 3 is 3

