CS & IT ENGINEERING

Algorithms

Dynamic Programming (DP)

Lecture No.- 04

By- Aditya Jain sir

Recap of Previous Lecture

Topic

Topic 0/1 KS

SOS

LCS

Topics to be Covered

Topic

MCM

TSP

MSG

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Topic : Dynamic Programming

Example:

Bottom-up Approach (Tabulation):

X = "ABCBDAB" → length = 7

Y = "BDCABA" → length = 6

Topic : Dynamic Programming

		B	D	C	A	B	A
	0	0	0	0	0	0	0
A	0	0	0	0	1	1	1
B	0	1	1	1	1	2	2
C	0	1	1	2	2	2	2
B	0	1	1	2	2	3	3
D	0	1	2	2	2	3	3
A	0	1	2	2	3	3	4
B	0	1	2	2	3	4	4

Length of LCS

Topic : Dynamic Programming

L[i–1, j–1]	L[i–1, j]
L[i, j–1]	1+ L[i, j]

Topic : Dynamic Programming

LCS Recurrence:

- $LCS(i, j) = 1 + LCS(i-1, j-1)$; if $X[i] = Y[j]$
- $LCS(i, j) = \max\{LCS(i, j-1), LCS(i-1, j)\}$; if $X[i] \neq Y[j]$

TD

BU

Topic : Dynamic Programming

Algorithm LCS based on Bottom-up Tabulation method :

Algorithm LCS (x, y)

Integer x[1..n], y [1..m];

```
{
        integer L [0..n , 0..m];
1.      for i ← 1 to n
                L [i, 0] = 0;
2.      for j ← 1 to m
                L [0 , j] = 0;
3.      for i ← 1 to n
                for j ← 1 to m
```

Topic : Dynamic Programming

```
if (x [i] = = y [j]) then
        L[i, j] = 1 + L [i – 1, j – 1];
else
        L[i, j] = max {L [i, j – 1], L [i – 1, j]};

  }
  return L(n,m)

}
```

TC: $O(n*m)$

SC: $O(n*m)$

MCM

Topic : Dynamic Programming

Matrix Chain Multiplication/Product (MCM):

#Q. Given two matrices A and B of size m×n and n×p.

How many scalar multiplication are reqruied in $A_{m\times n} * B_{n\times p} = C_{m\times p}$?

$\boxed{m * n * p}$

Topic : Dynamic Programming

Case 1:

Square Matrices:

$$A_{n\times n} * B_{n\times n} = C_{n\times n}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

Topic : Dynamic Programming

$c_{11} = a_{11} \times b_{11} + a_{12} \times b_{21}$ — 2

$c_{12} = a_{11} \times b_{12} + a_{12} \times b_{22}$ — 2

$c_{21} = a_{21} \times b_{11} + a_{22} \times b_{21}$ — 2

$c_{22} = a_{21} \times b_{12} + a_{22} \times b_{22}$ — 2

Every element = 2

Hence, total scalar multiplication = $2 \times \underbrace{(2 \times 2)} = 2 \times 2 \times 2 = 8$

Topic : Dynamic Programming

Case 2:

Non-Square Matrices:

$$A_{m\times n} * B_{n\times p} = C_{m\times p}$$

- Hence, for all m×p element.

  Total scalar multiplication = $n\times(m\times p) = m \times n \times p$

Important Result:

$$A_{m \times n} * B_{n \times p} = C_{m \times p}$$

- Total number of scalar multiplication = $\underbrace{m \times n \times p}$

Topic : Dynamic Programming

MCM Problem Statement:

- Given a chain of compatible non-square matrices, it is required to multiply them together to get a final resultant matrix., with min scalar multiplication

Topic : Dynamic Programming

Example:

$n = 3$

$[A_{2\times10}, B_{10\times50}, C_{50\times20}] = (ABC) = Z_{2\times20}$

1) How many?
2) min, max?

Topic : Dynamic Programming

Parenthesizing Problem:

ABC

(1) $(A * B) * C \Rightarrow \underline{3000}$

(2) $A * (B * C) \Rightarrow \underline{10,400}$

Topic : Dynamic Programming

(1) $(A * B) * C$

How many Scalar multiplication?

(i) $(A * B) = 2 \times 10 \times 50$

(ii) $(A * B) * C = 2 \times 50 \times 20$

So, total scalar multiplication for $(A * B) * C$ are

$$= 2 \times 10 \times 50 + 2 \times 50 \times 20$$

$$= 1000 + 2000 = 3000$$

Topic : Dynamic Programming

(2) $A * (B * C)$

(i) $(B * C) = 10 \times 50 \times 20$

(ii) $A * (B * C) = 2 \times 10 \times 20$

So, total scalar multiplication for $A * (B * C)$ are

$$= 10 \times 50 \times 20 + 2 \times 10 \times 20$$

$$= 10000 + 400$$

$$= 10400$$

MCM objective function

to minimize the number of scalar multiplication required

ABC

$A \times (B \times C)$ $(A \times B) \times C$

Topic : Dynamic Programming

DP based approach:-

- Given a chain of n matrixes (A1, A2.... An) where matrix Ai is of Dimension

  $P_{i-1} * P_i$.

  The MCM problem is to fully parenthesize the chain such that the total number of scalar multiplication are minimized.

A C

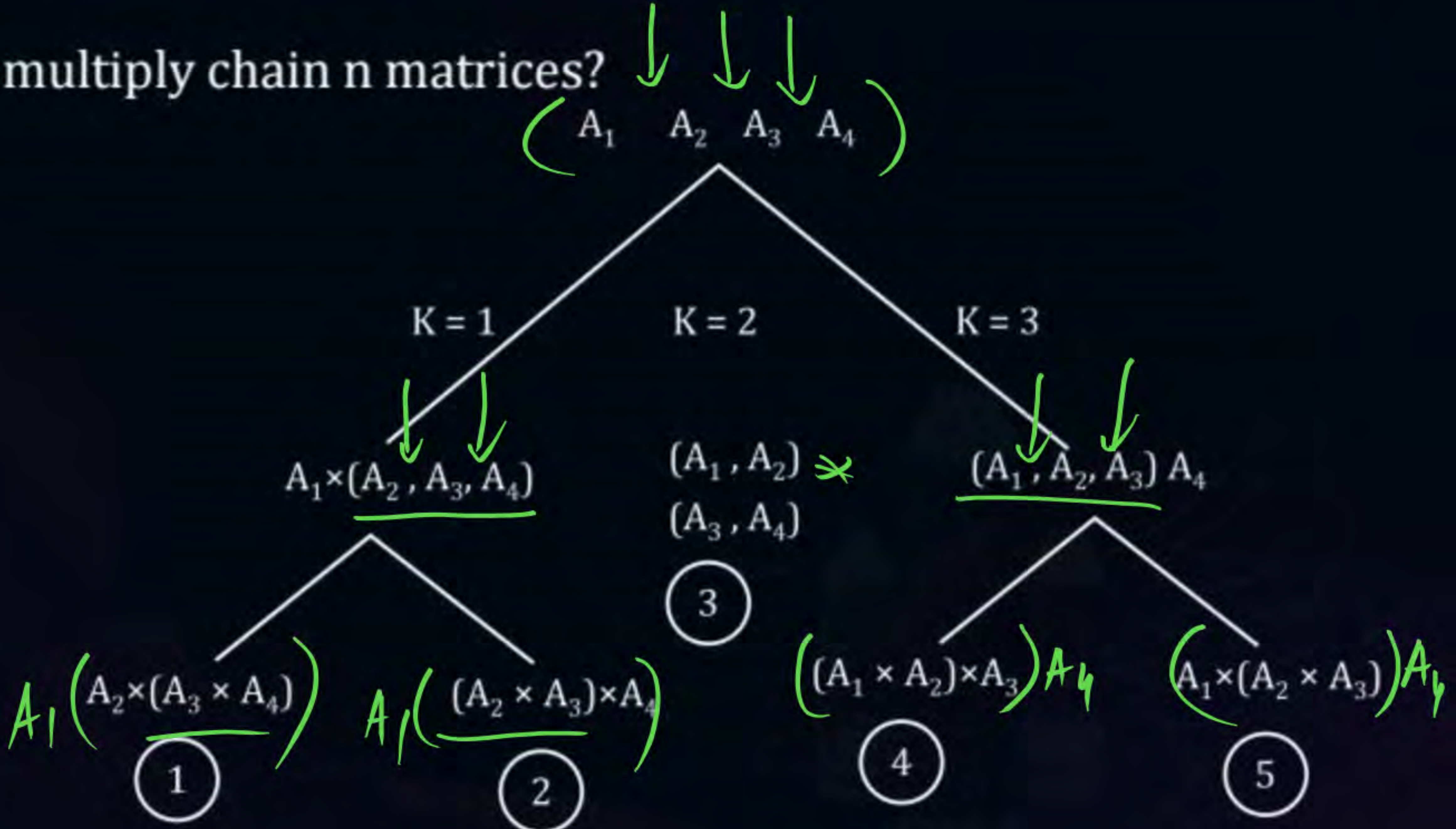
P [10, 2, 20, 30]

0 1 2 3

B

A1

#Q. How many ways to multiply chain n matrices?



5 ways

Topic : Dynamic Programming

Generalized Results:-

Total no. of ways to multiply a chain of (n + 1) matrix is ⇒ Catalan number (n)

n mat ⟶ Cat(n - 1)

E.g. N = ~~3~~ 4 → Catalan number = ~~2~~ 3

$$= \frac{1}{4} \times {}^{6}C_{3}$$

$$= \frac{1}{4} \times \frac{6 \times 5 \times 4 \times 3}{3! \times 3!}$$

$$= \frac{6 \times 5}{5} = 5$$

$$\boxed{\frac{1}{(n+1)} \times {}^{2n}C_{n}}$$

n +1

n ⇒ n-1

Topic : Dynamic Programming

Deamination of DP based Recurrence for MCM:

Let the resultant matrix Ai j be of product

$(A_i * A_{i+1} * A_{i+2} \ldots\ldots\ldots A_j)$

Chain → Result Ai j

$A_1 \times A_2 \times A_3 \ldots\ldots\ldots A_n \Rightarrow A_{1\times n}$ k = (1 to n–1)

Topic : Dynamic Programming

Any optimal Parameterization must split the chain about the matrix A_k & A_{k+1} such that the number of scalar multiplications are minimum.

Topic : Dynamic Programming

DP Recurrence

Let m[i, j] represent the min number of scalar multiplication to get the resultant matrix Aij.

$[A_i \times A_{i+1} \ldots\ldots\ldots A_k * \ldots\ldots A_j] \Rightarrow A\,i\,j \rightarrow M\,[i, j]$

Split at k

$\underbrace{(A_i \times A_i + \ldots\ldots A_k)}_{m\,[i\,,\,k]} \times \underbrace{(A_{k+1} \ldots\ldots\ldots A_j)}_{m\,[k+1,\,j]}$ Where $\rightarrow\ i \leq k \leq (j-1)$

Topic : Dynamic Programming

$$M[i, j] = \min\{m[i,k] + m[k+1, j] + P_{i-1} * P_k * P_j\}$$

V.imp

$i \le k \le (j-1)$

$(A_i \ldots\ldots\ldots\ldots A_k) * (A_{k+1} \ldots\ldots\ldots\ldots A_j)$

$P_{i-1} * P_i$ $\quad$ $P_{k-1} * P_k$

$P_{i-1} * P_k$

$P_k * P_{k+1}$ $\quad$ $P_{j-1} * P_j$

$P_k * P_j$

Topic : Dynamic Programming

eg

Example:

M[1,6]

$A_{16} = (A_1 \ A_2 \ A_3) \ (A_4 \ A_5 \ A_6)$

$(A_1 \ldots\ldots A_3)$ $(A_4 \ldots\ldots A_6)$

M [1,3] M [4,6]

2×3 $P_0\times P_1$ 5×8 $P_2\times P_3$ 8×4 $P_3\times P_4$ 7×6 $P_5\times P_6$

2×8 $P_0\times P_3$ 8×6 $P_3\times P_3$

Given,

$A_1 \rightarrow P_0 \times P_1 \rightarrow 2\times3$

$A_2 \rightarrow P_1 \times P_2 \rightarrow 3\times5$

$A_3 \rightarrow P_2 \times P_3 \rightarrow 5\times8$

$A_4 \rightarrow P_3 \times P_4 \rightarrow 8\times4$

$A_5 \rightarrow P_4 \times P_5 \rightarrow 4\times7$

$A_6 \rightarrow P_5 \times P_6 \rightarrow 7\times6$

: $[P_0 \ldots\ldots P_n]$

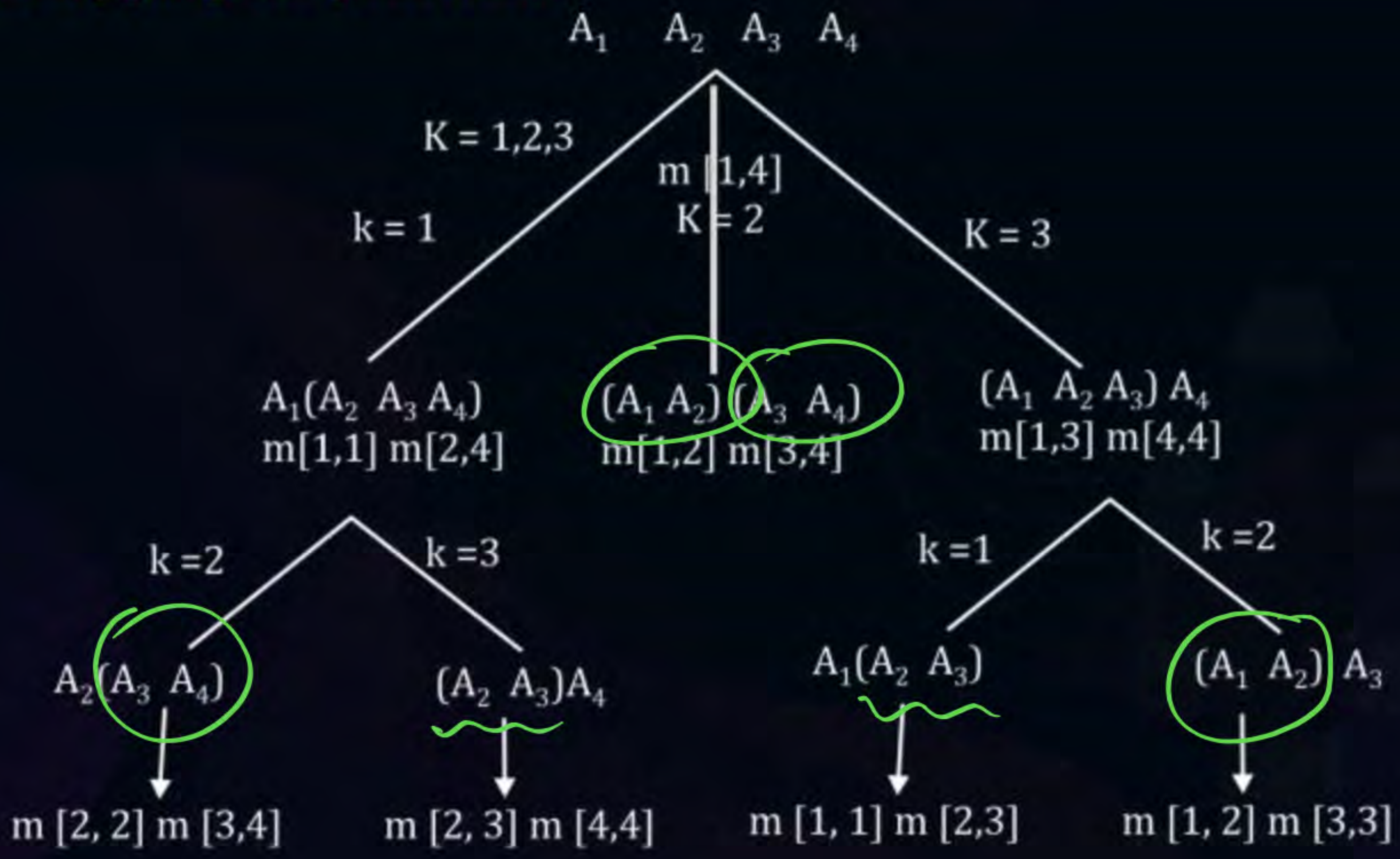
Number of both Resultants $= P_0 * P_3 \times P_6$

$= 2 \times 8 \times 6$

$i = 1, k = 3, j = 6$

Topic : Dynamic Programming

Are there overlapping Subproblems?



Topic : Dynamic Programming

Top Down: Recursion Approach

E.g. $A_1 A_2 A_3 A_4 = [2\times3, 3\times5, 5\times8, 8\times4]$

HW

$$\begin{array}{ccccc} P_0 & P_1 & P_2 & P_3 & P_4 \\ P:[2, & 3, & 5, & 8, & 4] \end{array}$$

Topic : Dynamic Programming

#Q. The minimum no. of scaler multiplication required to null The chain $A_1A_2A_3A_4$ = ?

Topic : Dynamic Programming :(DP)

Complexity Analysis of Top– down MCM (DP approach)

Assume n → no. of matrix

1. Time Complexity: $O(n^3)$ → for every length chain and for only every break-up point in it.

2. Space complexity:- $O(n^2)$

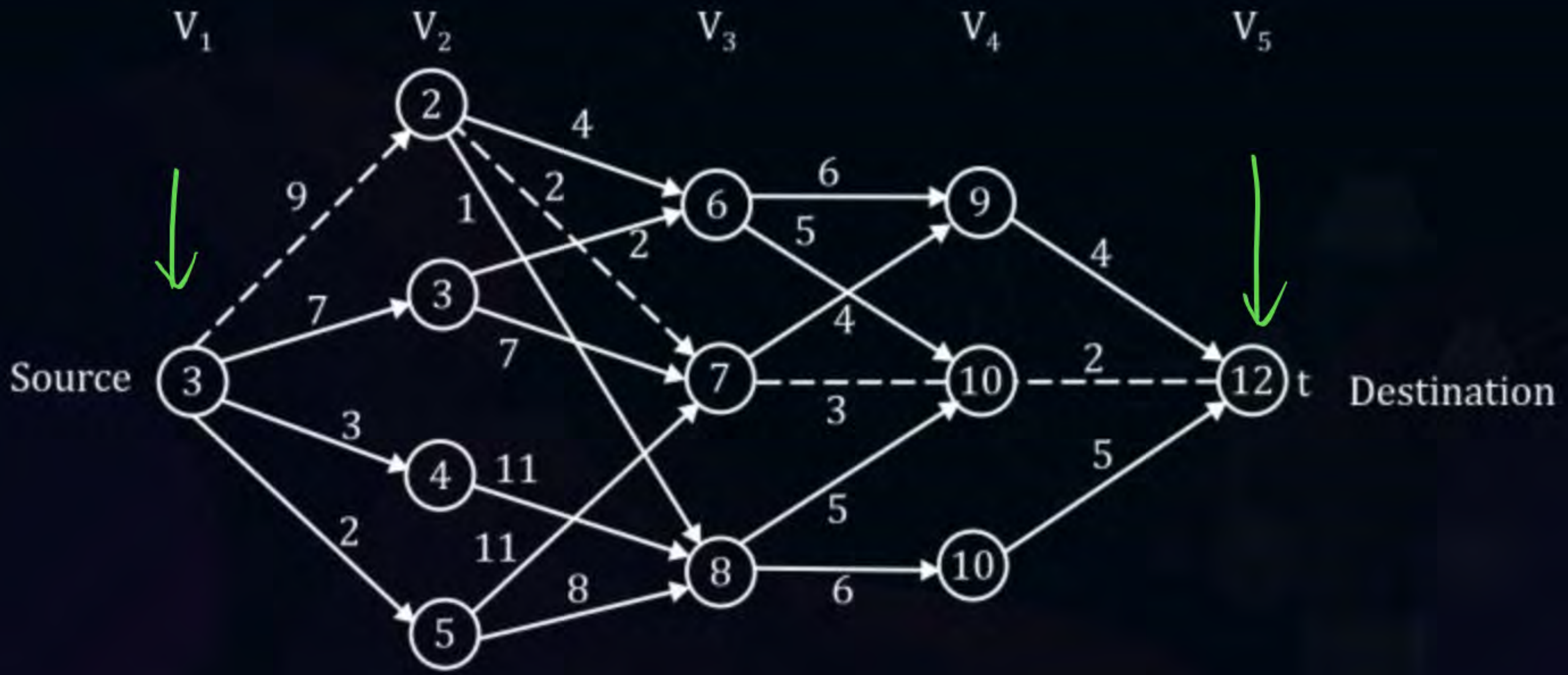
   Brute force / Enumeration: TC : $\Omega(2^n)$

MSG

TSP

Topic : Dynamic Programming :(DP)

7. Multi-Stage Graph:



Topic : Dynamic Programming :(DP)

Optimization Problem

Objective function

What criteria is to minimized/ maximized

(SPSP)

Minimize the cost of the path from source → Destination

Topic : Dynamic Programming :(DP)

Imp. Obs: n vertices , l Stage:

1. Source is always at stage 1
2. Destination is always at stage l (last stage)

3. An edge is always only from a stage to the next stage.

$$V_i \rightarrow V_{i+1}$$

Topic : Dynamic Programming :(DP)

1. $C(i,j)$: cost of edge from i^{th} vertex to j^{th} vertex

   $i \rightarrow j$

2. Cost (i,j) = cost of path from vertex 'j' present in stage 'i' to reach the destination vertex 'd'.

Topic : Dynamic Programming :(DP)

Multi-stage graph:

Source → Destination 1

Shortest path between 's' to 'd'

$= \min(\text{edge}(s \to k) + \text{Shortest path } (k \to d))$

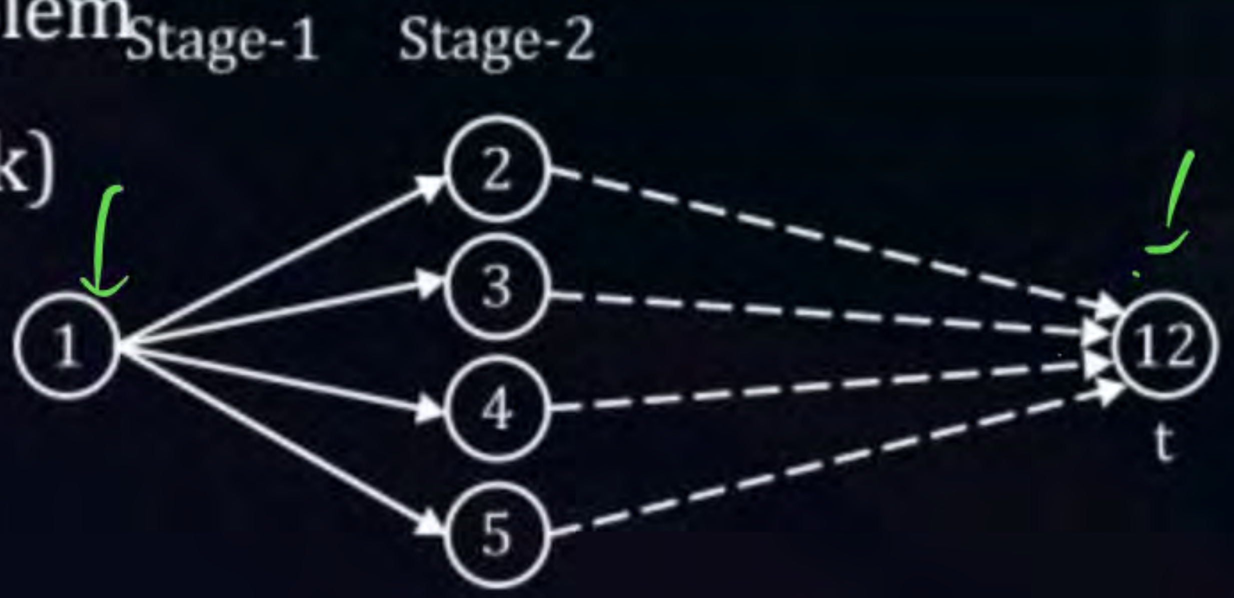
Recurrence

$\text{Cost}(1,1) = \min\{C(1,k) + \text{Cost}(2,k)\}$

Subproblem

$K \in V_2$ Cost (2, k)

(1, k)



and

Topic : Dynamic Programming :(DP)

Recurrence:

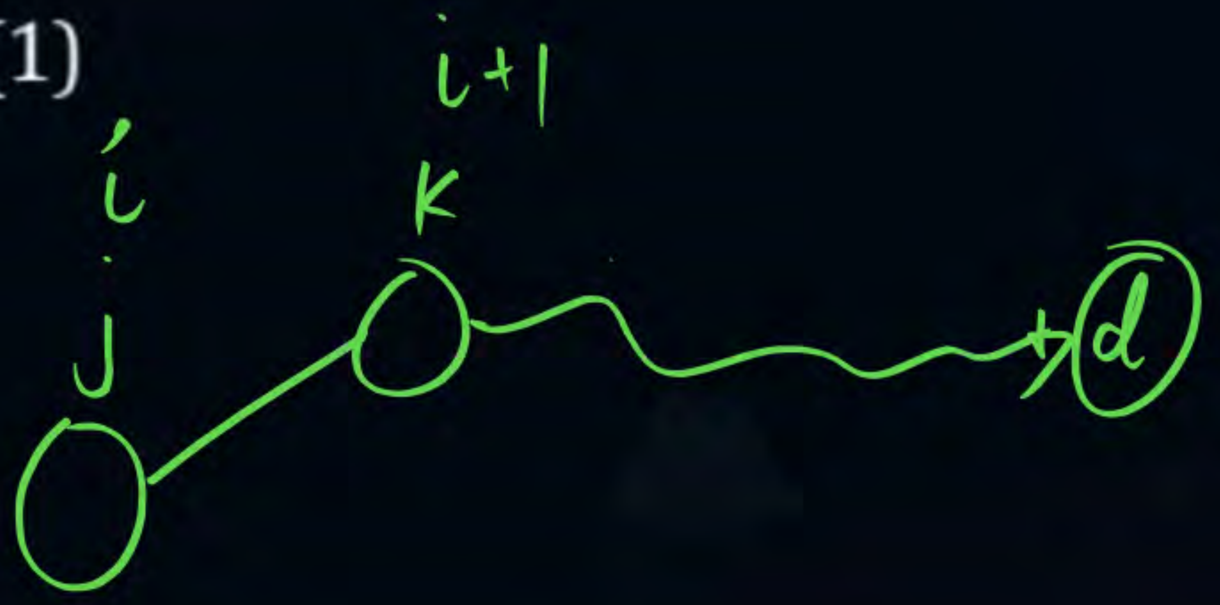
Cost (i, j) = Min {c (j, k) + cost (i +1, k)}....(1)

$K \in V_{i+1}$ and $(j, k) \in E$

l = total stages

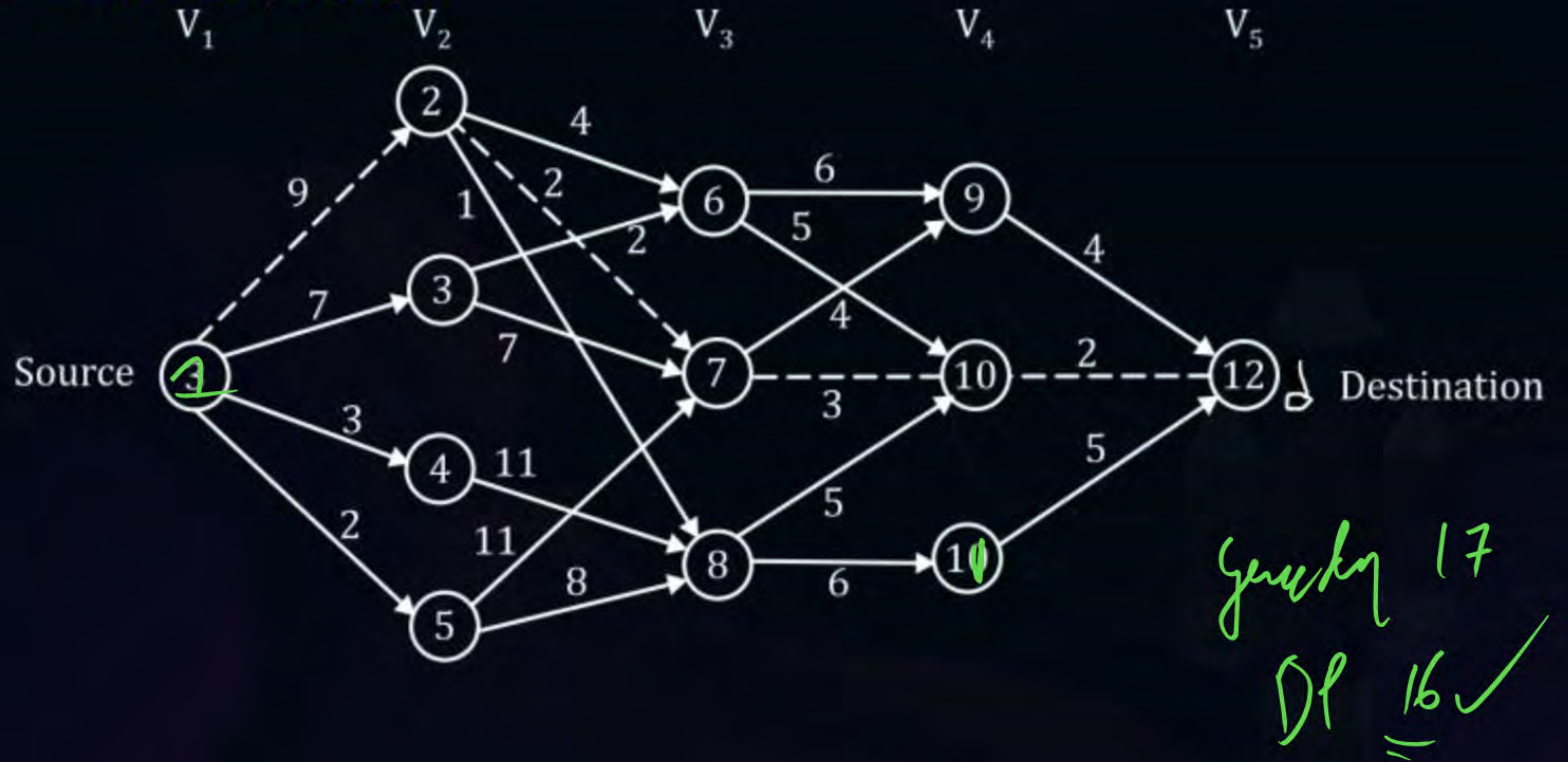
Cost (l -1, j) = c (j, d).........(2)

D(i, j) = 'k' that minimizes equation 1



Topic : Dynamic Programming :(DP)

E.g. 1. Multi- stage graph



Topic : Dynamic Programming :(DP)

Stage-1

$$\text{Cost}(1,1) = \min \left\{ \begin{array}{l} K = 2 : C(1,2) + \text{cost}(2,2) \\ K = 3 : C(1,3) + \text{cost}(2,3) \\ K = 4 : C(1,4) + \text{cost}(2,4) \\ K = 5 : C(1,5) + \text{cost}(2,5) \end{array} \right\}$$

Stage-1
↓
Stage-2
↓
Stage-3
↓
Stage-4

Topic : Dynamic Programming :(DP)

Stage-4 : (2^{nd} last stage)

$j = 9, 10, 11$, $k = 12$

$D(4, 9) = 12$

$Cost(4, 9) = c(9, 12) = 4$

$D(4, 10) = 12$

$Cost(4, 10) = c(10, 12) = 2$

$D(4, 11) = 12$

$Cost(4, 11) = c(11, 12) = 5$

Topic : Dynamic Programming :(DP)

Stage-3:

$j = 6,7,8$

Cost $(3, j)$

$$j=6: \text{Cost}(3, 6) = \text{Min}\begin{Bmatrix} K = 9: & C(6,9) + \text{Cost}(4, 9) \\ k = 10: & C(6, 10) + \text{Cost}(4, 10) \end{Bmatrix}$$

Prev Stage result

$$= \text{Min}\begin{Bmatrix} K = 9: & 6 + 4 \\ K = 10: & 5 + 2 \end{Bmatrix}$$

$$= \text{Min}\begin{Bmatrix} K = 9: & 10 \\ K = 10: & 7 \end{Bmatrix} \Rightarrow 7$$

$D(3, 6) = 10$

Topic : Dynamic Programming :(DP)

$$\text{Cost}(3,7) = \begin{Bmatrix} K = 9 : C(7,9) + \text{Cos}(4,9) \\ K = 10 : C(7,10) + \text{Cos}(4,10) \end{Bmatrix}$$

$$= \begin{Bmatrix} K = 9 : 4 + 4 \\ K = 10 ; 3 + 2 \end{Bmatrix}$$

$$= \text{Min} \begin{Bmatrix} K = 9 : \ 8 \\ K = 10 : 5 \end{Bmatrix} \Rightarrow 5$$

$D(3, 7) = 10$

Topic : Dynamic Programming :(DP)

$$\text{Cost}(3,8) = \text{Min}\begin{cases} K = 10: & C(8, 10) + \text{Cost}(4, 10) \\ K = 11: & c(8, 11) + \text{Cost}(4, 11) \end{cases}$$

$$= \text{Min}\begin{cases} K = 10: & 5 + 2 \\ K = 11: & 6 + 5 \end{cases}$$

$$= \text{Min}\begin{cases} K = 10: & 7 \\ K = 11: & 11 \end{cases} = 7$$

$D(3, 8) = 10$

Topic : Dynamic Programming :(DP)

Stage-2: $j = 2, 3, 4, 5$

$$\text{cost}(2,2) = \text{Min}\begin{Bmatrix} K=6: C(2,6) + \text{Cost}(3,6) \\ K=7: C(2,7) + \text{Cost}(3,7) \\ K=8: C(2,8) + \text{Cost}(3,8) \end{Bmatrix}$$

$$= \text{Min}\begin{Bmatrix} K=6: 4+7 \\ K=7: 2+5 \\ K=8: 1+7 \end{Bmatrix}$$

$$= \text{Min}\begin{Bmatrix} K=6: 11 \\ K=7: 7 \\ K=8: 8 \end{Bmatrix} = 7$$

$D(2,2) = 7$

Topic : Dynamic Programming :(DP)

$$j = 3$$
$$\text{Cost}(2,3) = \text{Min}\begin{Bmatrix} K = 6: C(3,6) + \text{Cost}(3,6) \\ K = 7: C(3,7) + \text{cost}(3,7) \end{Bmatrix}$$

$$\text{Cost}(2,3) = \text{Min}\begin{Bmatrix} K = 6: 2 + 7 \\ K = 7: 7 + 5 \end{Bmatrix} = \text{Min}(9, 12) = 9 \quad , D(2,3) = 6 \quad \text{Min}$$

$$j = 4$$
$$\text{Cost}(2,4) = \text{Min}\{K = 8: C(4,8) + \text{Cost}(3,8)\} = \text{Min}\{K = 8: 11 + 7\} = 18 \quad , D(2,4) = 8$$

$$j = 5$$
$$\text{Cost}(2,5) = \text{Min}\begin{Bmatrix} K = 7: 11 + 5 \\ K = 8: 8 + 7 \end{Bmatrix} \text{Min}(16, 15) = 15 \quad , D(2,5) = 8$$

Topic : Dynamic Programming :(DP)

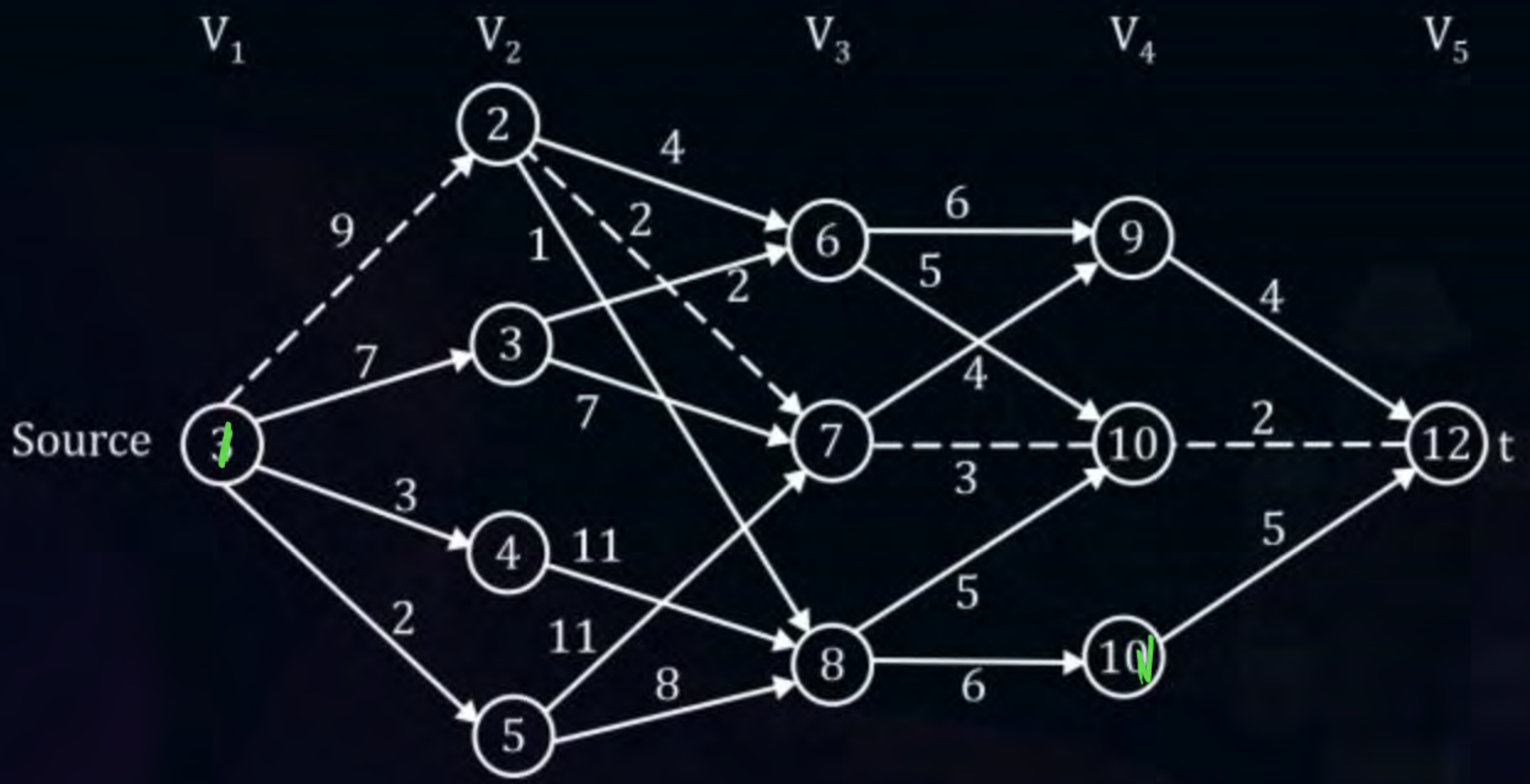
Stage-1

$$\text{Cost}(1,1) = \text{Min}\begin{Bmatrix} K = 2 : C\,(1,2) + \text{Cost}\,(2, 2) \\ K = 3 : C\,(1,3) + \text{Cost}\,(2, 3) \\ K = 4 : C\,(1,4) + \text{Cost}\,(2, 4) \\ K = 5 : C\,(1, 5) + \text{Cost}\,(2,5) \end{Bmatrix}$$

$$= \text{Min}\begin{Bmatrix} K = 2 : 9 + 7 \\ K = 3 : 7 + 9 \\ K = 4 : 3 + 18 \\ K = 5 : 2 + 15 \end{Bmatrix} = \text{Min}\,(16, 16, 21, 17) = 16 \quad , D(1,1) = 2 \text{ or } 3$$

#Q. What is the path that gives minimum cost?

E.g. 2.


V1
V2
V3
V4
V5
Source
t
9
7
3
2
4
2
1
2
7
11
11
8
6
5
4
3
5
6
4
2
5
1
2
3
4
5
6
7
8
9
10
10
12

Topic : Dynamic Programming :(DP)

8. Travelling Salesman Problem (TSP)

The tour of TSP should start from the home city (V_0) and visit remaining (n-1) cities exactly once and comeback to the home city (V_0), such that the cost of tour is minimize.

Home city = Starting vertex (V_0)

Tour: $V_0 \rightarrow$ n-1 (vertices) $\rightarrow V_0$

TSP $\rightarrow$ Optimization problem

Objective function $\rightarrow$ minimize the tour cost.

Topic : Dynamic Programming :(DP)

Cost adj. Matrix

C	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Greedy Method

$1 \to 2 \to 3 \to 4 \to 1$

$= 10 + 9 + 12 + 8$

$= 19 + 20$

$= 39$

Topic : Dynamic Programming :(DP)

The actual optimal/minimum tour cost is-

$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

$10 + 10 + 9 + 6$

$= 20 + 15$

$= 35$(Optimal solution)

Imp. Note:-

If we solve TSP by greedy, we get min tour cost as 39 ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$)

Which is not optimal solution.

Hence, greedy fails for TSP.

Topic : Dynamic Programming :(DP)

DP based solution:

Let g (i, s) represent the cost of the tour of TSP from vertex ‘i’ and visiting all remaining vertices in the set ‘s’ exactly once and then terminating the four at V_0 (source)

Let consider previous example $V_0 = 1$ as a starting vertex

E.g.

C	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Topic : Dynamic Programming :(DP)

$V_0 = 1$

$g(1, \{2, 3, 4\}) = \min [C(1, K) + g(K, S - \{k\})]$

$K \in S$ and $\langle 1,k \rangle \in E$

Original Problem $\rightarrow g(1, S)$

$[1 \rightarrow k \rightarrow 1]$

$C(l, k) + g(K, S - \{K\})$

↓

Subproblem

Topic : Dynamic Programming :(DP)

Generalized from → V. imp

$$g(i,s) = \min\, [c\,(i,k) + g\,(k,\, s - \{k\})] \quad \text{.......(1)}$$

$K \in S$

$\langle i, k \rangle \in E$

Starting vertex.

$i \rightarrow v_0$

$$\boxed{g(i, \varphi) = c\,(i, v_0)}$$

To make sure back to source v_0

$J\,(i, S)$ = Value of k that minimizes equation (1)

Topic : Dynamic Programming :(DP)

Bottom - up

E.g.

Cost adj. Matrix

C	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Original problem

$g(1, \{2,3,4,\}) =$

$$\text{Min}\left\{\begin{array}{l} K=2 : C(1,2) + \underline{g(2,\{3, 4\})} \\ K=3 : C(1,3) + \underline{g(3,\{2, 4\})} \\ K=4 : C(1,4) + \underline{g(4,\{2, 3\})} \end{array}\right\}$$

Topic : Dynamic Programming :(DP)

1. $|S| = 0,\ S = \varphi$

   $i = 2, 3, 4$

   $g(2, \varphi) = C\,(2, 1) = 5$, $J(2,\varphi) = 1$

   $g(3, \varphi) = C\,(3,1) = 6$, $J(3,\varphi) = 1$

   $g(4\ \varphi) = C\,(4,1) = 8$, $J(4,\varphi) = 1$

Topic : Dynamic Programming :(DP)

2. $|S| = 1$

HW

$$g(2, \{3\}) = C(2,3) + g(3, \varphi) \qquad , J(2,\{3\}) = 3$$

$$= 9 + 6 = 15$$

$$g(2, \{4\}) = C(2,4) + g(4, \varphi) = 10 + 8 = 18 \qquad , J(2,\{4\}) = 4$$

$$g(3, \{2\}) = C(3,2) + g(2, \varphi) = 13 + 5 = 18 \qquad , J(3,\{2\}) = 2$$

$$g(3, \{4\}) = C(3, 4) + g(4, \varphi) = 12 + 8 = 20 \qquad , J(3,\{4\}) = 4$$

Topic : Dynamic Programming :(DP)

$g(4, \{2\}) = C(4, 2) + g(2, \varphi)$, $J(4,\{2\}) = 2$

$= 8 + 5 = 13$

$g(4, \{3\}) = C(4,3) + g\{3, \varphi\}$

$= 9 + 6 = 15$, $J(4,\{3\}) = 3$

Topic : Dynamic Programming :(DP)

3. $|S| = 2$

$$g(2, \{3, 4\}) = \min \begin{Bmatrix} K = 3: C(2,3) + g(3, \{4\}) \\ K = 4: C(2,4) + g(4, \{3\}) \end{Bmatrix}$$

$$= \min \begin{Bmatrix} K = 3: 9 + 20 \\ K = 4: 10 + 15 \end{Bmatrix}$$ From prev. subproblem

$J(2, \{3,4\}) = 4$

$$= \min \begin{pmatrix} 29 \\ 25 \end{pmatrix} = 25$$

Topic : Dynamic Programming :(DP)

$$g(3, \{2, 4\}) = \min \begin{Bmatrix} K = 2: & C(3, 2) + g(2, \{4\}) \\ K = 4: & C(3, 4) + g(4, \{2\}) \end{Bmatrix}$$

$$= \min \begin{Bmatrix} K = 2: & 13 + 18 \\ K = 4 & 12 + 13 \end{Bmatrix}$$

$$J(3, \{2, 4\}) = 4$$

$$= \min \begin{pmatrix} 31 \\ 25 \end{pmatrix} = 25$$

Topic : Dynamic Programming :(DP)

$$g(4, \{2,3\}) = \min \begin{Bmatrix} K = 2: C(4,2) + g(2, \{3\}) \\ K = 3: C(4,3) + g(3, \{2\}) \end{Bmatrix}$$

$$= \min \begin{Bmatrix} K = 2: 8 + 15 \\ K = 3: 9 + 18 \end{Bmatrix}$$

$J(4, \{2, 3\}) = 2$

$$= \min \begin{Bmatrix} K = 2: 23 \\ K = 3: 27 \end{Bmatrix} = 23$$

Topic : Dynamic Programming :(DP)

Bottom - up

E.g.

Cost adj. Matrix

C	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Original problem

$$g(1,\{2,3,4,\}) = \text{Min}\left\{\begin{array}{l} K=2 : C(1,2) + g(2,\{3, 4\} \\ K=3 : C(1,3) + g(3,\{2, 4\} \\ K=4 : C(1,4) + g(4,\{2, 3\} \end{array}\right\}$$

Topic : Dynamic Programming :(DP)

$$g(1, \{2, 3, 4\}) = \min \begin{cases} K = 2: C(1,2) + g(2, \{3, 4\}) \\ K = 3: C(1,3) + g(3, \{2, 4\}) \\ K = 4: C(1,4) + g(4, \{2, 4\}) \end{cases}$$

$$= \min \begin{cases} K = 2: 10 + 25 \\ K = 3: 15 + 25 \\ K = 4: 20 + 23 \end{cases}$$

$$= \min \begin{cases} K = 2: 35 \\ K = 3: 40 \\ K = 4: 43 \end{cases} = 35$$

Topic : Dynamic Programming :(DP)

#Q. How to determine the path four which we got min cost?

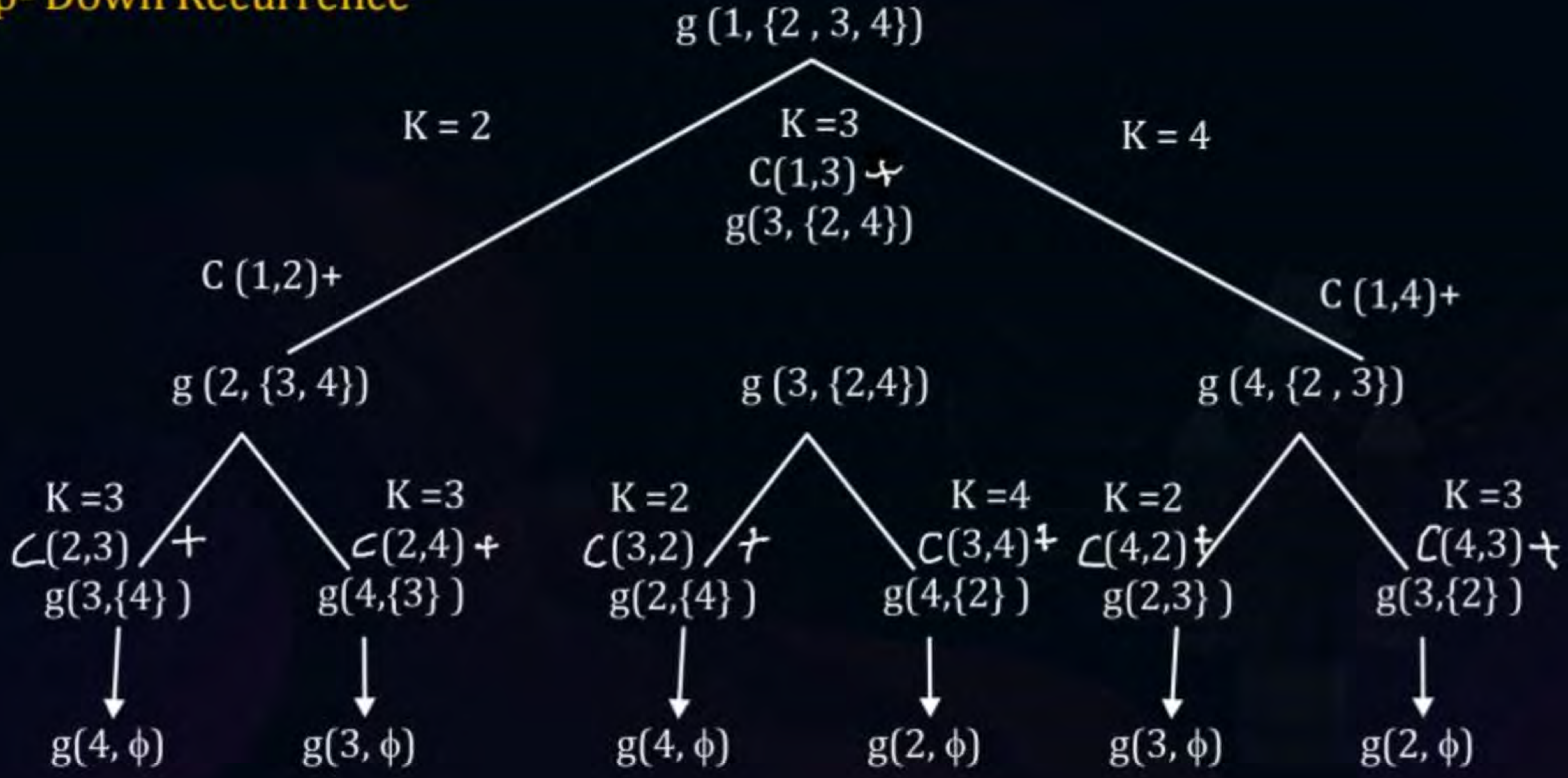
$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

Tour repeated by TSP

Cost = 35

Topic : Dynamic Programming :(DP)

Top- Down Recurrence



Topic : Dynamic Programming :(DP)

DP based Algo to solve the travelling salesman problem (TSP)

Remember

- Time complexity = $O(2^n * n^2)$
- Space complexity = $O(2^n * \sqrt{n})$, n = no. of cities.

Note : TSP is one of those problem for which there is no solution as of now which take polynomial Time complexity.

@ADITYASIR_PW

Telegram Link for Aditya Jain sir: https://t.me/AdityaSir_PW

THANK - YOU