DSA

Chapter ⑧ (graph)

~~2019~~ 2021

(i) Consider the following weight matrix W. (4)

$$W = \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix}$$

Draw weighted graph G. Apply modified Warshall's algorithm to find the shortest path Q.

(ii) A graph G is stored in memory as follows: (3)

	1	2	3	4	5	6	7	8
NODE	A	B		E		D	C	
NEXT	7	4	0	6	8	0	2	3
ADJ	1	2		5		7	9	

START = 1, AVAILN = 5

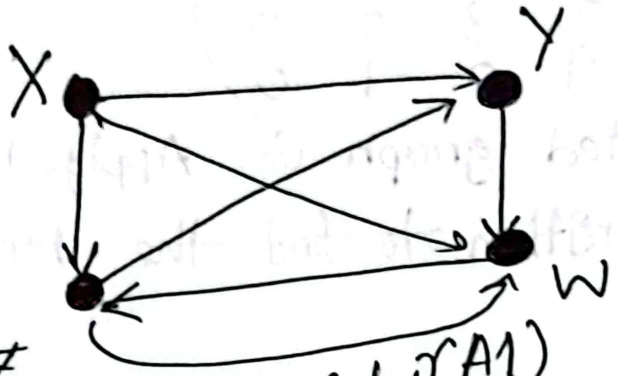
	1	2	3	4	5	6	7	8	9	10
DEST	2	6	4		6	7	4		4	6
LINK	10	3	6	0	0	0	0	4	0	0

AVAILE = 8

Draw the graph G

**(iii) Define complete graph and multigraph. (1)

Suppose the nodes of the figure A are stored in memory



Z

Is G strongly connected? (A1)

(1) Give the adjacency matrix A of the graph G) Calculate the path matrix P of G. ; find all the simple path from Y to Z ; find indeg (Y) and outdeg (Y). (3)

(iv) Consider the following figure. Find and print all the nodes reachable from the node A using DFS

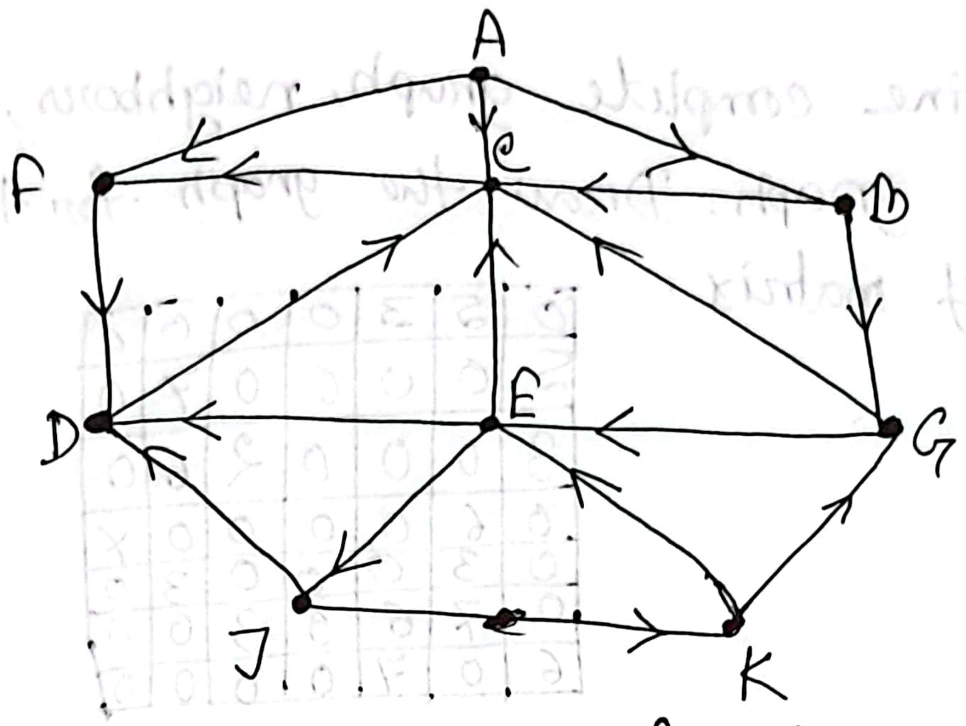


fig (A2)

2020

(i) algo to find the shortest path from a weighted graph (3)

(ii) Distinguish between BFS and DFS.

** (iii) consider fig A(2), find a minimum Path P From A to J using BFS where each edge has length 1.

Q A

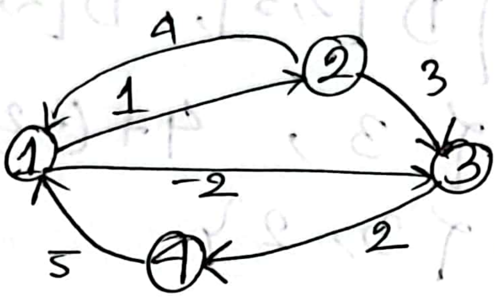
(19) (i) Define complete graph, neighbors, tree graph. Draw the graph for the adjacency matrix.

$$\begin{bmatrix} 0 & 5 & 3 & 0 & 0 & 0 & 6 \\ 5 & 0 & 0 & 6 & 0 & 7 & 0 \\ 3 & 0 & 0 & 0 & 2 & 6 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 7 \\ 0 & 3 & 0 & 0 & 0 & 3 & 0 \\ 0 & 7 & 6 & 0 & 3 & 0 & 0 \\ 6 & 0 & 7 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Chapter - 8
Graph

negative cycle মানে বুঝায় যদি edges এর weight যোগ করার পর negative মান আসে।
negative cycle graph G ~~এ~~ থাকবে না।

Floyd Warshell Algo for shortest path.



Distance matrix:

[Digonal zero হবে cause কোনো self loop নাই]

বৃত্তাচিহ্ন = vertex সংখ্যা

D^0

	1	2	3	4
1	0	1	-2	∞
2	4	0	3	∞
3	∞	∞	0	2
4	5	∞	∞	0

D^1

[D^0 এর প্রথম row colown same]

	1	2	3	4
1	0	1	-2	∞
2	4	0	2	∞
3	∞	∞	0	2
4	5	6	3	0

formula:

$$D^{K}[i,j] = \min\{D^{K-1}[i,j], D^{k-1}[i,K] + D^{k-1}[K,j]\}$$

$$D^{1}[2,3] = \min\{D^{1-1}[2,3], D^{1-1}[2,1] + D^{1-1}[1,3]\}$$

$$= \min\{D^{0}[2,3], D^{0}[2,1] + D^{0}[1,3]\}$$

$$= \min\{3, 4+(-2)\}$$

$$= \min\{3, 2\}$$

$$= 2$$

$$D^{1}[2,4] = \min\{D^{0}[2,4], D^{0}[2,1] + D^{0}[1,4]\}$$

$$= \min\{\alpha, 4+\alpha\}$$

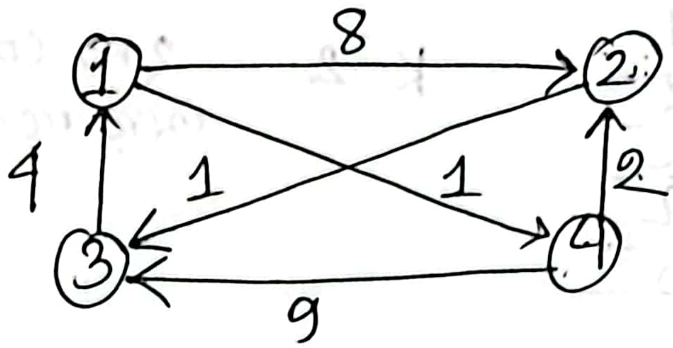
$$= \alpha$$

$$D^{1}[3,2] = \min\{D^{0}[3,2], D^{0}[3,1] + D^{0}[1,2]\}$$

$$= \min\{\alpha, \alpha+1\}$$

$$= \alpha$$

4



$$D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \alpha & 4 \\ 2 & \alpha & 0 & 1 & \alpha \\ 3 & 4 & \alpha & 0 & \alpha \\ 4 & \alpha & 2 & 9 & 0 \end{array}$$

$$D^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \alpha & 1 \\ 2 & \alpha & 0 & 1 & \alpha \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \alpha & 2 & 9 & 0 \end{array}$$

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 9 & 1 \\ 2 & \alpha & 0 & 1 & \alpha \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \alpha & 2 & 3 & 0 \end{array}$$

-3
-2 2-3

2-3→1

2-1 1-3
$\alpha + \alpha$

2-4-α

2-1-1-4
α+1

3-2→α

3-1-1-2
4+8=12

3-4→α

3-1-1-4
4+1

4-2→2

4-1-1-2→
α+8

$$D^3 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 8 & 9 & 1 \\ 2 & 5 & 0 & 1 & 6 \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \cancel{13}\ 7 & 2 & 3 & 0 \end{array}$$

1-2 = 8
1-3-3-2
1-4
1-3-3-4
2-1
2-3-3-1
2-4
2-3-3-4
4-1
4-3-3-1
4-2-2-3
4-2
4-3-3-2

2-4
2-3-3-4
1 3-1-4
4+1
= 1+5 = 6

$$D^4 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 & 1 \\ 2 & 5 & 0 & 1 & 6 \\ 3 & 4 & 7 & 0 & 5 \\ 4 & 7 & 2 & 3 & 0 \end{array}$$

1-2
1-4-4-2
1-3 → 9
1-4-4-3 ⇒ 10
2-4-4-2-3 ⇒ 4

3-1
3-4-4-1
3-2
3-4-4-2
3-1-4-2

2-1
2-4-4-1
2-3
2-4-4-3

⊛ Warshall's Algorithm for boolean path matrix P of the graph G.

Let G be a directed graph with m nodes, $v_1, v_2, \ldots v_m$.

P_K is the path matrix with ij entry is denoted by $P_K[i,j]$

$$P_K[i,j] = \begin{cases} 1 & \text{if there is a simple path from } v_i \text{ to } v_j \text{ which doesn't use any other nodes excepts } v_1, v_2, v_3 \ldots, v_K \\ 0 & \text{otherwise} \end{cases}$$

$P_0 = A$ the adjecency matrix of G.

G has m node so $P_m = P$ the path matrix of G.

$P_K[i,j] = 1$ can occur if

$P_{K-1}[i,j] = 1$ where is a simple path from v_i to v_j

[path matrix tells that is there any path between the nodes]

$P_{k-1}[i,k] = 1$ and $P_{k-1}[k,j] = 1$ a simple path from v_i to v_k, v_k to v_j

The elements of the matrix P_k can be obtained by

$$P_k[i,j] = P_{k-1}[i,j] \vee (P_{k-1}[I,K] \wedge P[K,J])$$

<u>Algo:</u> (1) Repeat for I, J = 1, 2, ... M: [Initialize P]

If A[I, J] = 0, then: Set P[I, J] := 0;

Else: Set P[I, J] := 1;

[End of loop]

(2) Repeat step 3 and 4 for K = 1, 2, .., M: [update P]

(3) Repeat step 4 for I = 1, 2, ... M:

(4) Repeat for J = 1, 2, ... M:

Set P[I, J] := P[I, J] ∨ (P[I, K] ∧ P[K, J])

[End of loop]

[End of loop step 3 loop]

[End of setp 2 loop] (5) Exit.

Shortest Path Algorithm:

Let G be a directed graph with m nodes, $v_1, v_2, \ldots, v_m$ suppose G is weighted. $w(e)$ = the weight or length of the edge e. Then G may be maintained in memory by its weight matrix $W = W(ij)$

$$W_{ij} = \begin{cases} w(e) & \text{if there is an edge from } v_i \text{ to } v_j \\ 0 & \text{if there is no edge from } v_i \text{ to } v_j \end{cases}$$

Q is the matrix that tells the shortest path V_i to V_j

$Q_K[i,j]$ = the smaller of the length of the preceding path from V_i to v_j or the sum of the lengths of the preceding paths from V_i to V_K and from V_K to V_j.

CamScanner

$Q_K[i,j] = MIN\left(Q_{K-1}[i,j],\ Q_{K-1}[i,K] + Q_{K-1}[K,j]\right.$

Algo: This algorithm finds a matrix Q such that Q[I,J] is the length of a shortest path from node V_I to node V_J. INFINITY is a very large number and MIN is the minimum value function.

1. Repeat for I, J = 1, 2, ... M: [Intialize Q]
   W[I,J] = 0, then: set Q[I,J] := INFINI
   Else: set Q[I,J] := W[I,J].
   [End of loop].
2. Repeat step 3 and 4 for K = 1, 2, ... M: [UPd
3. Repeat step 4 for I = 1, 2, ... M:
4. Repeat ~~step~~ for J = 1, 2, ... M:
   Set Q[I,J] := MIN (Q[I,J], Q[I,K] + Q[K
   [End of loop]
   [End of step 3 loop]
   [End of step 2 loop]
5. Exit

Exam 21

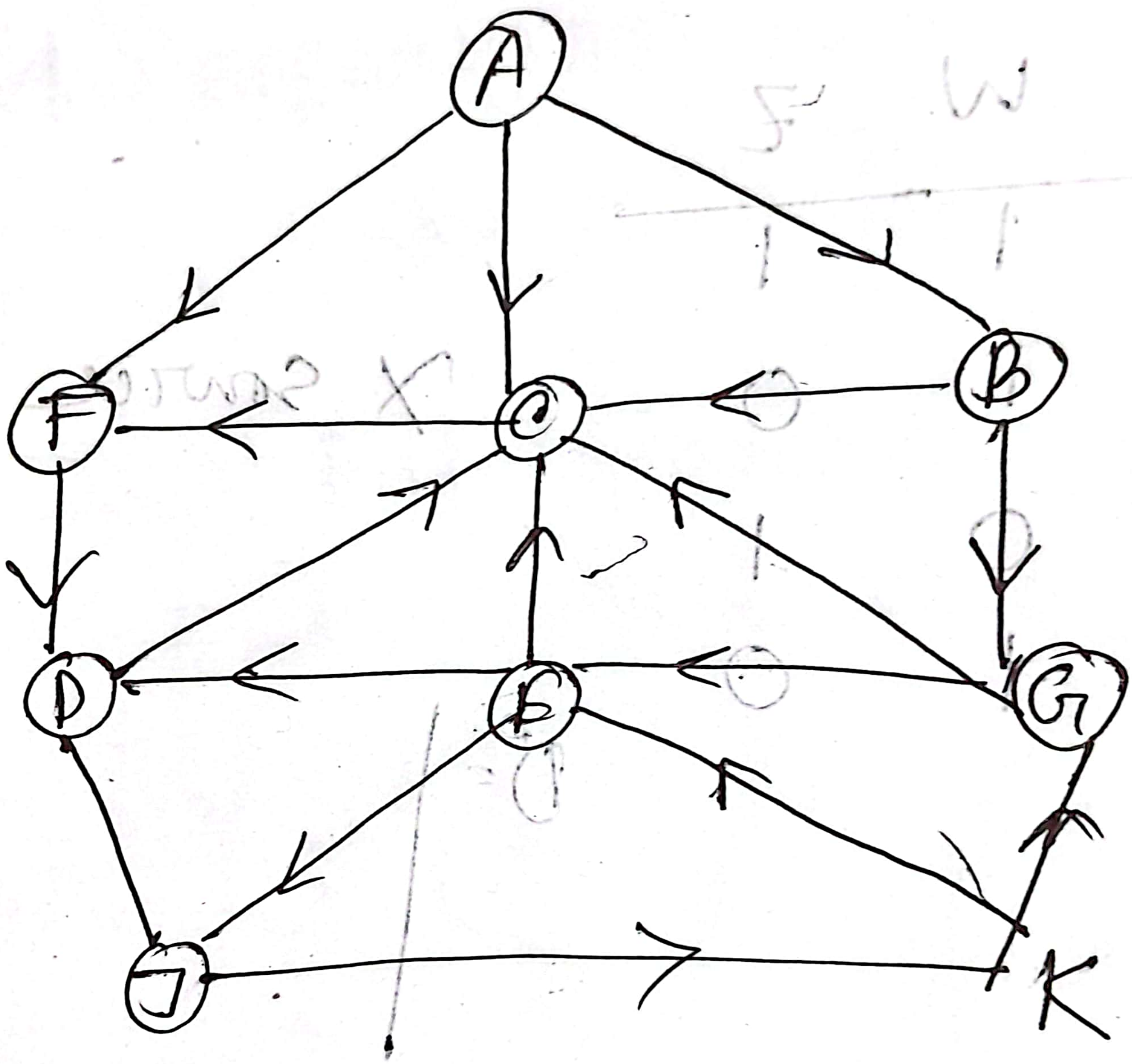
fig(A1) ~~adjecent matrix~~

adjacency lists
A : F, C, B
F : D
C : F
B : C, G
D : C,
E : C, D, J
G : C, E
J : K, D
K : G, E

প্রথমে node থেকে যেগুলো চলে যায় সেগুলো.

BFS: During the execution of the search, we will also keep track of the origin of each edge by using an array ORIG together with the array QUEUE.

① ~~In~~ first add A to Q. and add Null to orig in-first out



Adjacency list

A: F, C, B

F: D

C: E, G

A: F,

F: D

C: F

First in first out

Front

A	~~F C B~~

Rear

Front = 1   Q: A

Rear = 1   O: φ

② Remove A from Q by setting Front = front + 1 and Add A neighbors,

Front = 2   Q: A, F, C, B

Rear = 4   O: φ, A, A, A

~~A~~	F	C	B

③ Remove F.

F = 3   Q: A, F, C, B, D

R = 5   O: φ, A, A, A, F

~~A~~	~~F~~	C	B	D

④ Remove C.

F = 4   Q: A, F, C, B, D

R = 5   O: φ, A, A, A, F

its neighbour F is already in the Queue.

~~A~~	~~F~~	~~C~~	B	D

CamScanner

(5) Remove B :

~~A~~	~~F~~	~~C~~	~~B~~	D	G

F = 5
R = 6

Q = A, F, C, B, D, ~~G~~, G
O = φ, A, A, ~~A, F~~, F, B

(6) Remove D :

~~A~~	~~F~~	~~C~~	~~B~~	~~D~~	G

F = 6
R = 6

Q = A, F, C, B, D, G
O = φ, A, A, A, F, B

(7) Remove G :

~~A~~	~~F~~	~~C~~	~~B~~	~~D~~	~~G~~	E

F = 7
R = 7

Q = A, F, C, B, D, G, E
O = φ, A, A, A, F, B, G

(8) Remove E :

~~A~~	~~F~~	~~C~~	~~B~~	~~D~~	~~G~~	~~E~~	J

F = 8
R = 8

Q = A, F, C, B, D, G, E, J
O = φ, A, A, A, F, B, G, E

(9) Remove J :

~~A~~	~~F~~	~~C~~	~~B~~	~~D~~	~~G~~	~~E~~	~~J~~	K

F = 9
R = 9

Q = A, F, C, B, D, G, E, J, K
O = φ, A, A, A, F, B, G, E, J

to reach J to A we will use backtrack using the array orig.

(10) J ← E ← G ← B ← A is the required path.

Stack → (last in first out.) Exam 20

DFS : প্রথম node থেকে শুরু করে এটার উপরে push করি A

① push A : Stack : J

② pop A print top element A and then push onto the stack all the neighours of A

Print: A | Stack: F, C, B
↳ last

③ pop B ; print B | Stack: F, C, G

④ pop G , print G | Stack: F, C, E

⑤ pop E , print E | Stack: F, C, D, J

(6) pop J, print ~~E~~ J | stack: F, C, D, K

(7) pop K, print K | stack: F, C, D,

(8) pop D, print D | stack: F, C

(9) pop C, print C | stack: F

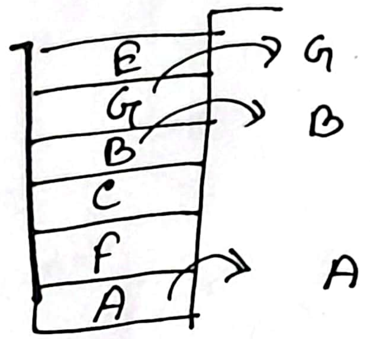
(10) pop F, print F | stack:

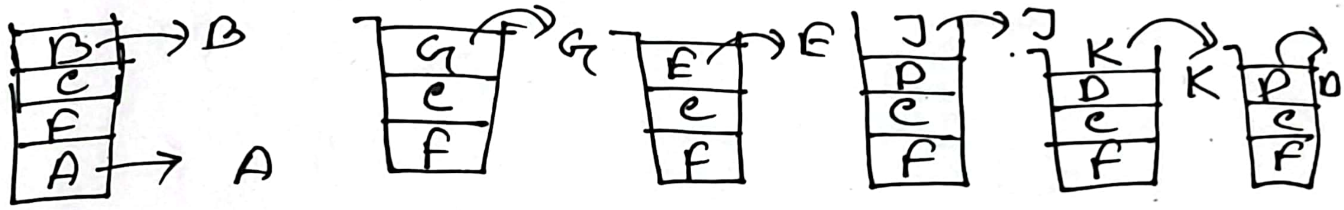
The stack is empty now. so the D.F.S of G starting at node A is now complete

A, B, G, E, J, K, D, C, F

these nodes are reachable from J.

A F. D J. E. K , G, B

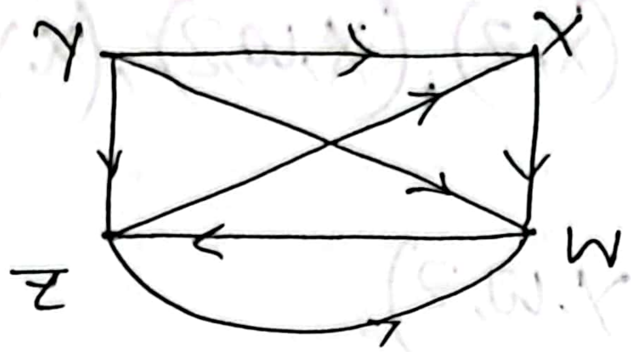




A B G E J K D C F

Exam 21

[1]



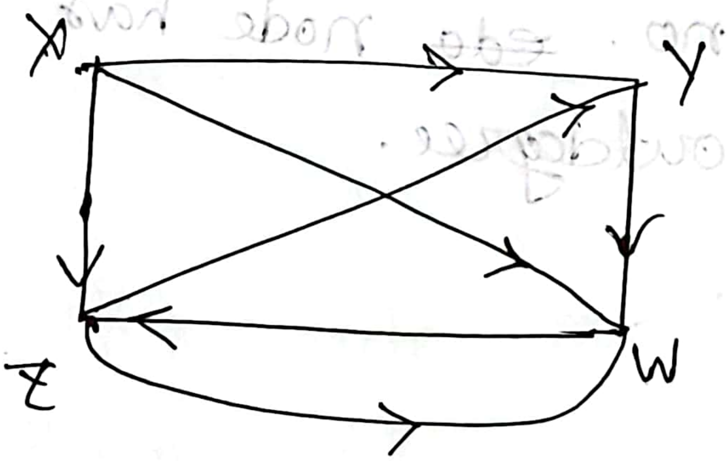
(X Y Z W)

Adjecency matrix A =

$$\begin{array}{c|cccc} & X & Y & Z & W \\ \hline X & 0 & 0 & 0 & 1 \\ Y & 1 & 0 & 1 & 1 \\ Z & 1 & 0 & 0 & 1 \\ W & 0 & 0 & 1 & 0 \end{array}$$

(a) All the simple paths from X to Z

Ans:



A =

$$\begin{array}{c|cccc} & X & Y & W & Z \\ \hline X & 0 & 1 & 1 & 1 \\ Y & 0 & 0 & 1 & 0 \\ W & 0 & 0 & 0 & 1 \\ Z & 0 & 1 & 1 & 0 \end{array}$$

X to Z = (X, Z), (X, W, Z), (X, Y, W, Z)

(b) Y to Z = (Y, W, Z),

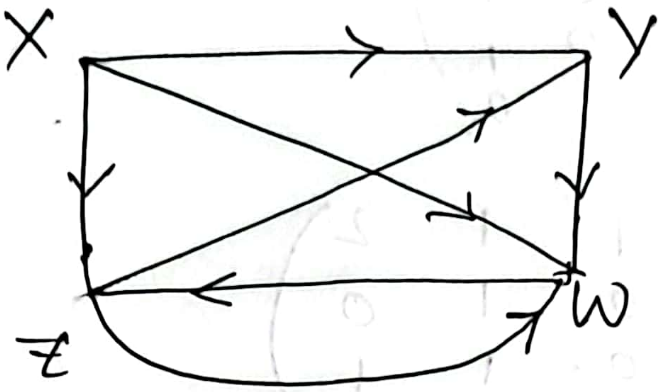
(c) Indeg(Y) = 2 দুটি আসবে
Outdg(Y) = 1

(d) source X Indeg = 0
outdeg = 3

(e) sink no. ~~edo~~ node has a
nonzero outdegree.

Exam 21

1. (X, Y, Z, W)



(a) Find the adjacency matrix A of the graph G.

$V_1 = X, \quad V_2 = Y, \quad V_3 = Z, \quad V_4 = W.$

$$A = \begin{array}{c c} & \begin{array}{cccc} X & Y & Z & W \end{array} \\ \begin{array}{c} X \\ Y \\ Z \\ W \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

(b) G has node $m = 4$.

$B_4 = A + A^2 + A^3 + A^4$

path matrix

$B_4 = P$ [সব মান 0,1 দ্বারা repla

[illegible] বেগম

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$B_4 = A + A^2 + A^3 + A^4$$

$$B_4 = \begin{pmatrix} 0 & 5 & 6 & 8 \\ 0 & 1 & 2 & 3 \\ 0 & 3 & 3 & 5 \\ 0 & 2 & 3 & 3 \end{pmatrix}$$

The path matrix is now obtain by setting $P_{ij} = 1$ wherever there is a nonzero entry in the matrix B_4

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

(c) G is not strongly connected because there is no path to X from any node.

Exam-24

(8.8) $A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

using warshall's algo for path matrix.

$P^0 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$P^1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$P^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$$P_3 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$P_4 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$