EXPERIMENT - 1 12/1/23 Write a program for distance vector algorithm for find suitable path for transmission - Bellman ford. # include < stolio. h> # include < stallib. h> int Bellman-Ford (int G[20][20] int V int E, int edge [20][2]) int 1, u, v, k, distance [20], pourent [20], s or (i=0; i<x; i++) D'distance [i] = 1000 parent [i] = -1; perint ("Enter cource:"). ceauf ("% d", &s); distance [s-1]=0; for (i=0; i < v-1; i++). for (k=0; k<E; k++) y = edge [k][o], v = edge [k][i];
if (distance [v] + G [u][v] <

distance [v]) distance [v] = distance [n] +G[n](v) parent [v] = u; for (k=0; k<E; k++) u=edge[k][o], V=edge[k][l];
if (distance[u] + g[u][v] < distance[x]
flag=0;

Chang parent [i] + 1);

parent = %d \n', i+1, distance [i]

parent [i] + 1);

neturn flag; int main U int V, edge [20][2], 6[20][20], q 1, j, k=0; print ("BELLMAN FORD \n"); print ("Enter no. of vertices:") print ("Enter graph in matrin form): \n"). scalif ("rd", sv); for (i= 0; i=V, i++) 9 for (j=0; j<V; j+t) scand ("r.d", eq[i][j]); if (G[i][j]!=0) edge [k][o]=i, edge [k++][l]=j; if (Bellnan Ford (G, V, l, edge))

print (" \n No regative weight eycle \n");

else print (" \n Negative weight eycle

exist \n"). netway 0;

DUTPUT: BELLMAN FORD Enter no. of vertices: 5 Enter graph in natrin form: D 2 2 99 99 2 0 99 3 99 2 99 0 6 4 99 3 6 5 5 99 99 4 5 0 Enter coura: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3 No negative weight cycle.			PAGE NO:
BELLMAN FORD Enter no. of vertices: 5 Enter graph in natrin form: 0 2 2 99 99 2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter cource: 1 Vertin 1 -> coet = 0 parent = 0 Vertin 2 -> coet = 2 parent = 1 Vertin 3 -> coet = 5 parent = 2 Vertin 5 -> coet = 6 parent 3			
BELLMAN FORD Enter no. of vertices: 5 Enter graph in natrin form: 0 2 2 99 99 2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter cource: 1 Vertin 1 -> coet = 0 parent = 0 Vertin 2 -> coet = 2 parent = 1 Vertin 3 -> coet = 5 parent = 2 Vertin 5 -> coet = 6 parent 3			1 1 2
Enter po of vertices: 5 Enter graph in natrin form: 0 2 2 99 99 2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3		OUTPUI:	
Enter po of vertices: 5 Enter graph in natrin form: 0 2 2 99 99 2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3			
D 2 2 99 99 2 D 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 8 parent 3	Į.	BELLMAN FORD	
D 2 2 99 99 2 D 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 8 parent 3		Enter no. of vertices: 5	
D 2 2 99 99 2 D 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 8 parent 3		Enter graph in natrin fe	orm:
2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter cource: 1 Vertin 1 -> coet = 0 parent = 0 Vertin 2 -> coet = 2 parent = 1 Vertin 3 -> coet = 2 parent = 1 Vertin 4 -> coet = 5 parent = 2 Vertin 5 -> coet = 6 parent 3		CA S ARROWS	
2 0 99 3 99 2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter cource: 1 Vertin 1 -> coet = 0 parent = 0 Vertin 2 -> coet = 2 parent = 1 Vertin 3 -> coet = 2 parent = 1 Vertin 4 -> coet = 5 parent = 2 Vertin 5 -> coet = 6 parent 3		D 2 2 99 99	15. N
2 99 0 6 4 99 3 6 0 5 99 99 4 5 0 Enter cource: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3			
99 3 6 0 5 99 99 4 5 0 Enter source: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 8 parent 3			
Enter source: 1 Vertin 1 -> cost = 0 pagent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3		00 2 0	38
Enter cource: 1 Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3	· C &	90 0000	1
Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3		7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	
Vertin 1 -> cost = 0 parent = 0 Vertin 2 -> cost = 2 parent = 1 Vertin 3 -> cost = 2 parent = 1 Vertin 4 -> cost = 5 parent = 2 Vertin 5 -> cost = 6 parent 3		E. A	DO PERC
Vertin 2 -> cool = 2 parent = 1 Vertin 3 -> cool = 2 parent = 1 Vertin 4 -> cool = 5 parent = 2 Vertin 5 -> cost = 6 parent 3		When course !	t in NV
Verter 4 -> cost = 5 parent = 1 Verter 5 -> cost = 6 parent 3		Verlin 1 -> cost = 0 parent = t	
Verter 4 -> cost = 5 parent = 1 Verter 5 -> cost = 6 parent 3	3	Verler 2 -> Coct = 2 parent	-
Vertin 5 -> cost = 5 parent = 2			
No negative weight cycle.		Vertex 4 -> col = 5 parent	= 2
No negative weight cycle.		Vertin 5 -> cost = 8 page t	. 3
No negative weight cycle.		pv en	
		No regalise princht and	
		Jest cycle.	
	-	1,20	
	R 1-1		
	3		
	· UA'	The state of the s	