12/1/23

EXPERIMENT - 11

Write a program for distance vector algorithm to find suitable path for transmission - Bellman ford.

```
#include <stdio.h>
#include <stdlib.h>
int Bellman_Ford (int G[20][20], int V,
 int E, int edge [20][2])
{
  int i, u, v, k, distance [20], parent [20], s,
  flag = 1;
    for (i=0; i<V; i++)
      distance [i] = 1000, parent [i] = -1;
      printf ("Enter source :");
      scanf ("%d", &s);
       distance [s-1] = 0;
      for (i=0; i < V-1 ; i++)
        {
          for (k=0; k<E; k++)
            {
              u = edge [k][0], v = edge [k][1];
             if (distance [u] + G [u][v] <
                  distance [v])
             distance [v] = distance [u] + G[u][v],
              parent [v] = u;
            }
        }
    for (k=0; k<E; k++)
      {
      u = edge [k] [0], v= edge [k] [1];
     if (distance[u] + G[u][v] < distance[v])
      } flag = 0;
```

```
if (flag)
    for (i=0; i<V; i++)
        printf ("Vertex %d → cost = %d
    parent = %d \n", i+1, distance [i],
            parent [i] + 1);
    return flag;
}

int main()
{
 int V, edge [20][2], G[20][20], i, j, k=0;
 printf ("BELLMAN FORD \n");
 printf ("Enter no. of vertices:");
  scanf ("%d", &V);
    printf ("Enter graph in matrix
    form: \n");
      for (i=0; i<V; i++)
        for (j=0; j<V; j++)
        {
          scanf ("%d", &G[i][j]);
          if (G[i][j] != 0)
              edge [k][0] = i, edge [k++][1]=j;
        }
if (Bellman_Ford (G, V, k, edge))
        printf ("\n No negative weight cycle \n");
else printf ("\n Negative weight cycle
              exists \n");
      return 0;
    }
```

PTO

PAGE NO:
DATE:

OUTPUT:

BELLMAN FORD
Enter no. of vertices: 5
Enter graph in matrix 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
Vertex 1 → cost = 0 parent = 0
Vertex 2 → cost = 2 parent = 1
Vertex 3 → cost = 2 parent = 1
Vertex 4 → cost = 5 parent = 2
Vertex 5 → cost = 6 parent 3

No negative weight cycle.

21/1/2023