12-01-22

6) Write a program for distance vector algorithm to find suitable path for transmission

```
#include <stdio.h>
#include <stdlib.h>

int Bellman_Ford(int G[20][20], int v, int i,
                 int edge[20][20])
{
 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;
}
```

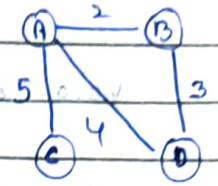
```
if (flag)
    for (i=0; i<v; i++)
        printf("Vertex %d -> cost = %d  parent = %d
                \n", i+1, distance[i], parent[i]+1);

    return flag;
}

int main()
{
    int v, edge[20][2], G[20][20], i, j, k=0;
    Printf("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][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;
}
```

Output:

Enter no of vertices: 4
Enter graph in matrix form:



0 2 5 4
2 0 999 3
5 999 0 999
4 3 999 0

Enter source: 1

Vertex 1 → cost = 0 parent = 0
Vertex 2 → cost = 2 parent = 1
Vertex 3 → cost = 5 parent = 1
Vertex 4 → cost = 4 parent = 1

No negative weight cycle