

12/01/2023

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

DATE

PAGE

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

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

```
int BellmanFord (int G[20][20], int V, int  
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;
}

```



D/P

Enter the no. of vertices: 4

Enter graph in matrix form:

0 5 17 3

2 0 3 5

8 5 0 2

1 3 2 0

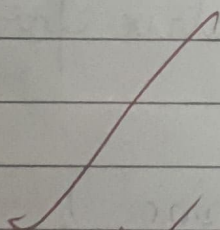
Enter source: 1

Vertex 1  $\rightarrow$  cost = 0 parent = 0

Vertex 2  $\rightarrow$  cost = 5 parent = 1

Vertex 3  $\rightarrow$  cost = 5 parent = 4

Vertex 4  $\rightarrow$  cost = 3 parent = 1

  
M/1/23