Program Assignment 3

Aim : Distance Vector Routing Algorithm using Bellman Ford's Algorithm.

Code :

```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);//Enter the nodes
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost
matrix
            rt[i].from[j]=j;
        }
    }
        do
        {
            count=0;
            for(i=0;i<nodes;i++)//We choose arbitary vertex k and we calculate the
direct distance from the node i to k using the cost matrix
            //and add the distance from k to node j
            for(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
                if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                {//We calculate the minimum distance
                    rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                    rt[i].from[j]=k;
                    count++;
```

```
                }
        }while(count!=0);
        for(i=0;i<nodes;i++)
        {
            printf("\n\n For router %d\n",i+1);
            for(j=0;j<nodes;j++)
            {
                printf("\t\nnode %d via %d Distance %d
",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
        }
    printf("\n\n");
    getch();
}
//reference
//https://www.thelearningpoint.net/computer-science/c-program-distance-vector-routi
ng-algorithm-using-bellman-ford-s-algorithm
```

Output :

```
E:\sem 6\cn-tuts\BFA-RIP.exe

Enter the number of nodes : 3

Enter the cost matrix :
0 2 7
2 0 1
7 1 0


 For router 1

node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 2 Distance 3

 For router 2

node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1

 For router 3

node 1 via 2 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0
```

Conclusion:
RIP was implemented successfully.

Program 2

Aim : OPEN SHORTEST PATH FIRST ROUTING PROTOCOL

Code :

```
#include <stdio.h>
#include <string.h>
int main()
{
    int count, src_router, i, j, k, w, v, min;
    int cost_matrix[100][100], dist[100], last[100];
    int flag[100];
    printf("\n Enter the no of routers");
    scanf("%d", &count);
    printf("\n Enter the cost matrix values:");
    for (i = 0; i < count; i++)
    {
        for (j = 0; j < count; j++)
        {
            printf("\n%d->%d: ", i, j);
            scanf("%d", &cost_matrix[i][j]);
            if (cost_matrix[i][j] < 0)
                cost_matrix[i][j] = 1000;
        }
    }
    printf("\n Enter the source router: ");
    scanf("%d", &src_router);
    for (v = 0; v < count; v++)
    {
        flag[v] = 0;
        last[v] = src_router;
        dist[v] = cost_matrix[src_router][v];
    }
    flag[src_router] = 1;
    for (i = 0; i < count; i++)
    {
        min = 1000;
        for (w = 0; w < count; w++)
        {
            if (!flag[w])
                if (dist[w] < min)
                {
                    v = w;
                    min = dist[w];
```

```
                }
        }
        flag[v] = 1;
        for (w = 0; w < count; w++)
        {
            if (!flag[w])
                if (min + cost_matrix[v][w] < dist[w])
                {
                    dist[w] = min + cost_matrix[v][w];
                    last[w] = v;
                }
        }
    }
    for (i = 0; i < count; i++)
    {
        printf("\n%d==>%d: Path taken: %d", src_router, i, i);
        w = i;
        while (w != src_router)
        {
            printf("\n<--%d", last[w]);
            w = last[w];
        }
        printf("\n Shortest path cost: %d", dist[i]);
    }
}
// reference
//https://forgetcode.com/C/1368-OPEN-SHORTEST-PATH-FIRST-ROUTING-PROTOCOL
```

Output :

```
E:\sem 6\cn-tuts\ospf.exe

 Enter the no of routers2

 Enter the cost matrix values:
0->0: 3

0->1: 4

1->0: 5

1->1: 6

 Enter the source router: 1

1==>0: Path taken: 0
<--1
 Shortest path cost: 5
1==>1: Path taken: 1
 Shortest path cost: 6
--------------------------------
Process exited after 16.93 seconds with return value 2
Press any key to continue . . .
```

Conclusion : OSPF was implemented successfully.