DISTANCE VECTOR ROUTING PROTOCOL

Aim

To implement and simulate algorithm for distance vector routing protocol.

Theory

Distance - Vector Routing Protocol in data networks is basically used to determine the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass through, where one router counts as one hop. In some cases, distance-vector protocols also take into account network latency and other factors that influence traffic on a given route. To determine the best route across a network routers, on which a distance vector protocol is implemented, exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information.

Distance-vector routing protocols also require that a router informs its neighbours of network topology changes periodically. Distance-vector routing protocols commonly use the Bellman–Ford algorithm and Ford–Fulkerson algorithm to calculate the best possible route. The term distance vector refers to the fact that the protocol manipulates vectors of distances to other nodes in the network. In the earlier days, this routing algorithm was implemented more widely in local area networks with the Routing Information Protocol(RIP).

Algorithm

Bellman-Ford Algorithm

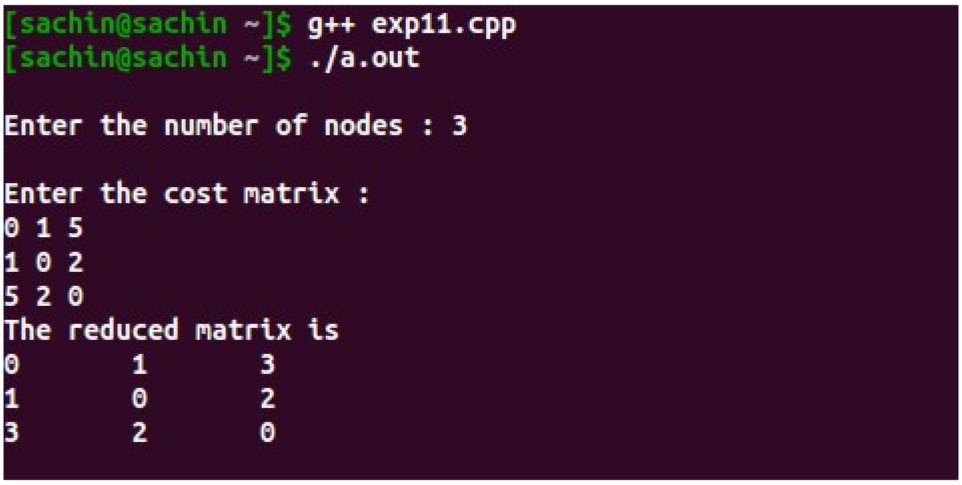
```
1. START
2. START PROCEDURE BellmanFord
3. Input the size(n) and elements of the element matrix
4. FOR i = 0 t o n DO
5.   FOR j = 0 t o n DO
6.        FOR k = 0 t o n DO
7.             dist[i][j]=min(dist[i][j],arr[i][k]+ dist[k][j])
8.        END FOR
9.   END FOR
10.END FOR
11.END PROCEDURE BellmanFord
12.STOP
```

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);
 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];
   rt[i].from[j]=j;
  }
 }
 do
 {
  count=0;for(i=0;i<nodes;i++)
  for(j=0;j<nodes;j++)
  for(k=0;k<nodes;k++)
  if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
  {
   rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
   rt[i].from[j]=k;
   count++;
  }
 }
 while(count!=0);
 printf("The reduced matrix is");
 for(i=0;i<nodes;i++)
 {
  printf("\n");
  for(j=0;j<nodes;j++)
  {
   printf("%d\t",rt[i].dist[j]);
  }
 }
 printf("\n\n");
}
```

Output



```
[sachin@sachin ~]$ g++ exp11.cpp
[sachin@sachin ~]$ ./a.out

Enter the number of nodes : 3

Enter the cost matrix :
0 1 5
1 0 2
5 2 0
The reduced matrix is
0       1       3
1       0       2
3       2       0
```

Result

implemented and simulateed algorithm for distance vector routing protocol in c. The output was obtained and verified.