PROGRAM 7

- ASHISH KUMAR
- 2K18/SE/041

Aim:- Write a C++ program to implement link state routing algorithm.

Theory:- Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.
Link state routing is the second family of routing protocols. While distance vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.
Features of link state routing protocols –
• Link state packet – A small packet that contains routing information.
• Link state database – A collection information gathered from link state packet.
• Shortest path first algorithm (Dijkstra algorithm) – A calculation performed on the database results into shortest path.

To find shortest path, each node need to run the famous Dijkstra algorithm. This famous algorithm uses the following steps:
• Step-1: The node is taken and chosen as a root node of the tree, this creates the tree with a single node, and now set the total cost of each node to some value based on the information in Link State Database
• Step-2: Now the node selects one node, among all the nodes not in the tree like structure, which is nearest to the root, and adds this to the tree. The shape of the tree gets changed.
• Step-3: After this node is added to the tree, the cost of all the nodes not in the tree needs to be updated because the paths may have been changed.
• Step-4: The node repeats the Step 2. and Step 3. until all the nodes are added in the tree.

CODE:-

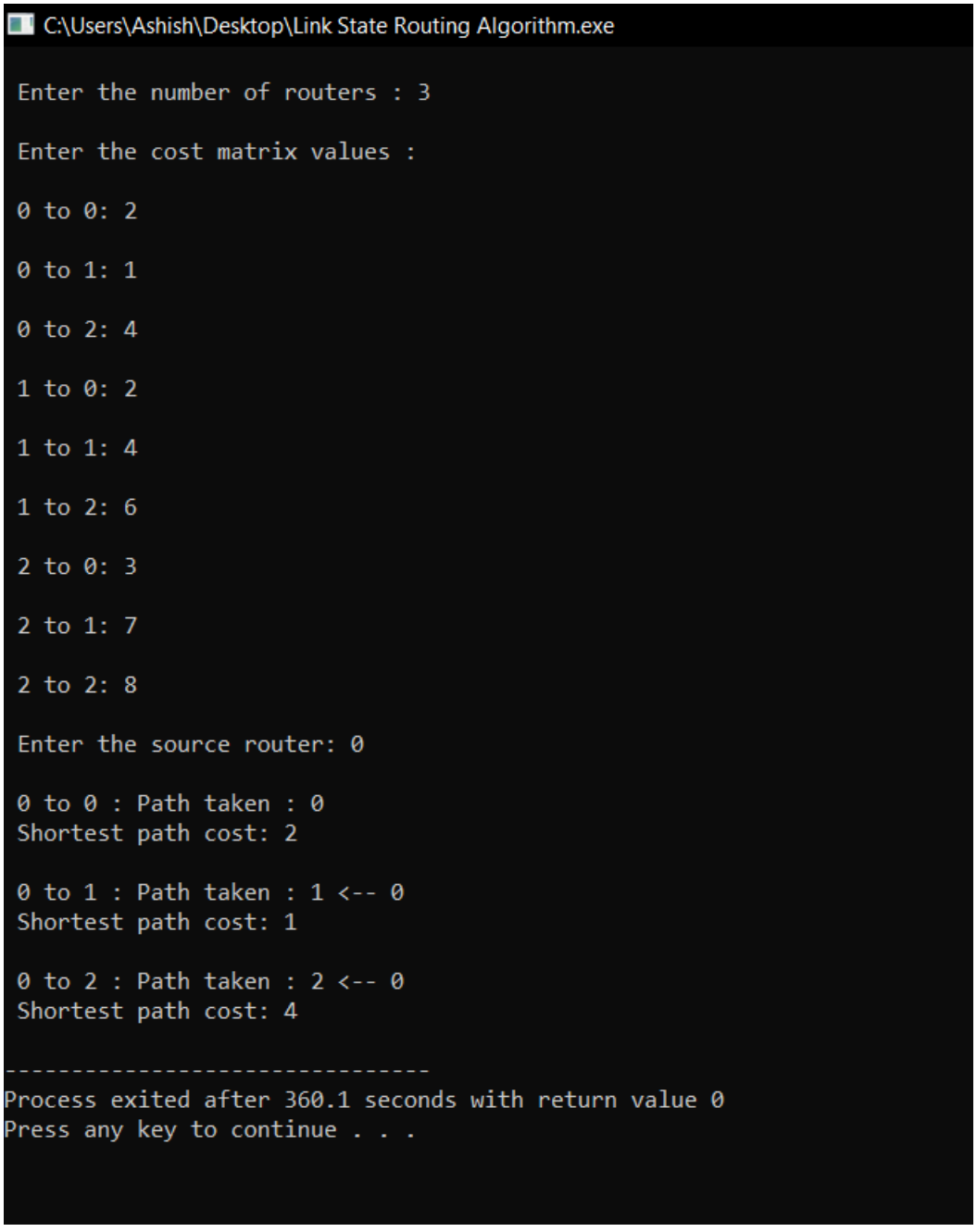
```
#include<iostream>

#include<bits/stdc++.h>

#include <string.h>
```

```
using namespace std;
int main()
{
int count,src_router,i,j,k,w,v,n;
int cost_matrix[100][100],dist[100],last[100],min;
int flag[100];
cout<<"\n Enter the number of routers : ";
cin>>count;
cout<<"\n Enter the cost matrix values : \n";
for(i=0;i<count;i++)
{
for(j=0;j<count;j++)
{
cout<<" \n "<<i<<" to "<<j<<": ";
cin>>cost_matrix[i][j];
if(cost_matrix[i][j]<0)cost_matrix[i][j]=1000;
}
}
cout<<"\n Enter the source router: ";
cin>>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++)
{
cout<<"\n "<<src_router<<" to "<<i<<" : Path taken : "<<i;
w=i;
while(w!=src_router)
{
cout<<" <-- "<<last[w];
w=last[w];
}
cout<<"\n Shortest path cost: " <<dist[i]<< "\n";
}
return 0;
}
```

OUTPUT:-


C:\Users\Ashish\Desktop\Link State Routing Algorithm.exe
Enter the number of routers : 3
Enter the cost matrix values :
0 to 0: 2
0 to 1: 1
0 to 2: 4
1 to 0: 2
1 to 1: 4
1 to 2: 6
2 to 0: 3
2 to 1: 7
2 to 2: 8
Enter the source router: 0
0 to 0 : Path taken : 0
Shortest path cost: 2
0 to 1 : Path taken : 1 <-- 0
Shortest path cost: 1
0 to 2 : Path taken : 2 <-- 0
Shortest path cost: 4

Process exited after 360.1 seconds with return value 0
Press any key to continue . . .

Finding & Learning:- We have learned and understand the concept of link state protocol and found this :

Link State protocols in comparison to Distance Vector protocols have:

1. It requires large amount of memory.
2. Shortest path computations require many CPU circles.
3. If network use the little bandwidth; it quickly reacts to topology changes
4. All items in the database must be sent to neighbors to form link state packets.
5. All neighbors must be trusted in the topology.