**Batch:A2**

**Roll Number: 16010421063** **Experiment Number:08**

**Name:Arya Nair**

**Title of the Experiment:Routing algorithm**

---

**Program:**

```
#include <stdio.h>


int main()
{
int min,i,j,nv,current,source,dest,T,y,x,v,c;
int max=1000;
int hi=1;
int visited[50],path[50],distance[50],adj[50][50];
 printf("Enter the number of vertices in the graph:\n");
 scanf("%d",&v);
 printf("Enter the adjacnecy matrix:\n");
 for(i=1;i<=v;i++)
   for(j=1;j<=v;j++)
     scanf("%d",&adj[i][j]);


 for(i=1;i<=v;i++)
 {
   distance[i]=max;
   visited[i]=0;
   path[i]=0;
 }
```

```
printf("Enter the source vertex:\n");

scanf("%d",&source);



current=source;

visited[current]=1;

nv=1;

T=0;



while(nv!=v)

   for(i=1;i<=v;i++)

  {

  if(adj[current][i]!=0)

      if(visited[i]!=hi)

      if(distance[i]>adj[current][i]+T)

      {

          distance[i]=adj[current][i] +T;

          path[i]=current;



      }



  }

  min=max;

  for(i=1;i<=v;i++)

      if(visited[i]!=hi)

      if(distance[i]<min)

      {
```

```
            min=distance[i];
            current =i;
        }
        visited[current]=1;
        nv++;
        T=distance[current];
    }
    printf("Enter the destination vertex:\n");
    scanf("%d",&dest);
    printf("Shortest distance from source to destination is
%d\n",distance[dest]);
    printf("Shortest path between source %d and destination %d
is:\n",source,dest);
    y=dest;

    do{
        i=2;
        x=path[y];
        printf("%d --> %d\n",x,y);
        y=x;
    }while(y!=source);



return 0;
}
```

**Output:**

```
→  EXP 8 git:(main) ✗ ./a.out
Enter the number of vertices in the graph:
6
Enter the adjacnecy matrix:
20 25 26 27 29 21
30 35 38 32 34 39
40 45 48 42 44 49
50 55 58 52 54 59
60 65 68 62 64 69
70 75 78 72 74 79
Enter the source vertex:
2
Enter the destination vertex:
5
Shortest distance from source to destination is 34
Shortest path between source 2 and destination 5 is:
2 --> 5
→  EXP 8 git:(main) ✗
```

**Post Lab Question- Answers (If Any):**

**1) The shortest path in routing can refer to**

a) The least expensive path

b) The least distant path

c) The path with the smallest number of hops

d) Any or a combination of the above

**Ans- D**

**2) In Distance Vector Routing each router receives vectors from**

a) Every router in the network

b) Every router less than two units away

c) A table stored by the software

d) Its neighbors only

**Ans- A**

**3) Link State routing is a _____routing algorithm**

a) Static

b) Dynamic

c) Both

d) Any

**Ans- B**

**4) In the network layer the packet is frequently called as _____**

a) Message

b) Frame

c) Datagram

d) None of the Above

**Ans- C**

**5) What is Traffic Shaping?**

**Ans.** Traffic shaping (also known as packet shaping) is bandwidth management technique that delays the flow of certain types of network packets in order to ensure network performance for higher priority applications. Traffic shaping essentially limits the amount of bandwidth that can be consumed by certain types of applications. It is primarily used to ensure a high quality of service for business-related network traffic. The most common type of traffic shaping is application-based traffic shaping. Fingerprinting tools are first used to identify the application associated with a data packet. Based on this, specific traffic shaping policies are applied. For example, you might want to use application-based traffic-shaping to throttle peer-to-peer file sharing, while giving maximum bandwidth to a business-critical application such as Voice-over-IP (VoIP), which is especially sensitive to latency

CO3. Build the skills of sub-netting and routing mechanisms.
CO4. Execute their knowledge of computer communication principles, including Error detection and correction, multiplexing, flow control, and error control.

**Conclusion:** Understood and implemented the Dijkstra's Algorithm in python