**Batch: B1**

**Roll Number: 1914078 Experiment Number: 7**

**Name: Devansh Shah**

**Title of the Experiment: Implementation of Routing Algorithm**

**Program:**

#include<stdio.h>

#define MAX 20

#define INFINITY 99999

int adj[MAX][MAX]; /\*Adjacency matrix \*/

int n; /\* Denotes number of nodes in the graph \*/

void dijkstra(int adj[MAX][MAX],int n,int startnode);

int main()

{

int max\_edges,i,j,origin,destin,d,u;

char graph\_type;

printf("Enter number of nodes : ");

scanf("%d",&n);

max\_edges=n\*(n-1);

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

{

for(j=1;j<=n;j++)

{

adj[i][j]=99999;

}

}

for(i=1;i<=max\_edges;i++)

{

printf("Enter edge %d( 0 0 to quit )source and destination : ",i);

scanf("%d %d",&origin,&destin);

if( (origin==0) && (destin==0) )

break;

if( origin > n || destin > n || origin<=0 || destin<=0)

{

printf("Invalid edge!\n");

i--;

}

else

{

printf("Enter distance:");

scanf("%d",&d);

adj[origin][destin]=d;

adj[destin][origin]=d;

}

}/\*End of for\*/

printf("The adjacency matrix is :\n");

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

{

for(j=1;j<=n;j++)

{ if (i==j)

adj[i][j]=0;

printf("%6d",adj[i][j]);

}

printf("\n");

}

printf("\nEnter the starting node:");

scanf("%d",&u);

dijkstra(adj,n,u);

}/\*End of main()\*/

void dijkstra(int adj[MAX][MAX],int n,int startnode)

{

int cost[MAX][MAX],distance[MAX],pred[MAX];

int visited[MAX],count,mindistance,nextnode,i,j;

//pred[] stores the predecessor of each node

//count gives the number of nodes seen so far

//create the cost matrix

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

for(j=1;j<=n;j++)

if(adj[i][j]==0)

cost[i][j]=INFINITY;

else

cost[i][j]=adj[i][j];

//initialize pred[],distance[] and visited[]

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

{

distance[i]=cost[startnode][i];

pred[i]=startnode;

visited[i]=0;

}

distance[startnode]=0;

visited[startnode]=1;

count=1;

while(count<n-1)

{

mindistance=INFINITY;

//nextnode gives the node at minimum distance

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

if(distance[i]<mindistance&&!visited[i])

{

mindistance=distance[i];

nextnode=i;

}

//check if a better path exists through nextnode

visited[nextnode]=1;

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

if(!visited[i])

if(mindistance+cost[nextnode][i]<distance[i])

{

distance[i]=mindistance+cost[nextnode][i];

pred[i]=nextnode;

}

count++;

}

//print the path and distance of each node

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

if(i!=startnode)

{

printf("\nDistance of node%d=%d",i,distance[i]);

printf("\nPath=%d",i);

j=i;

do

{

j=pred[j];

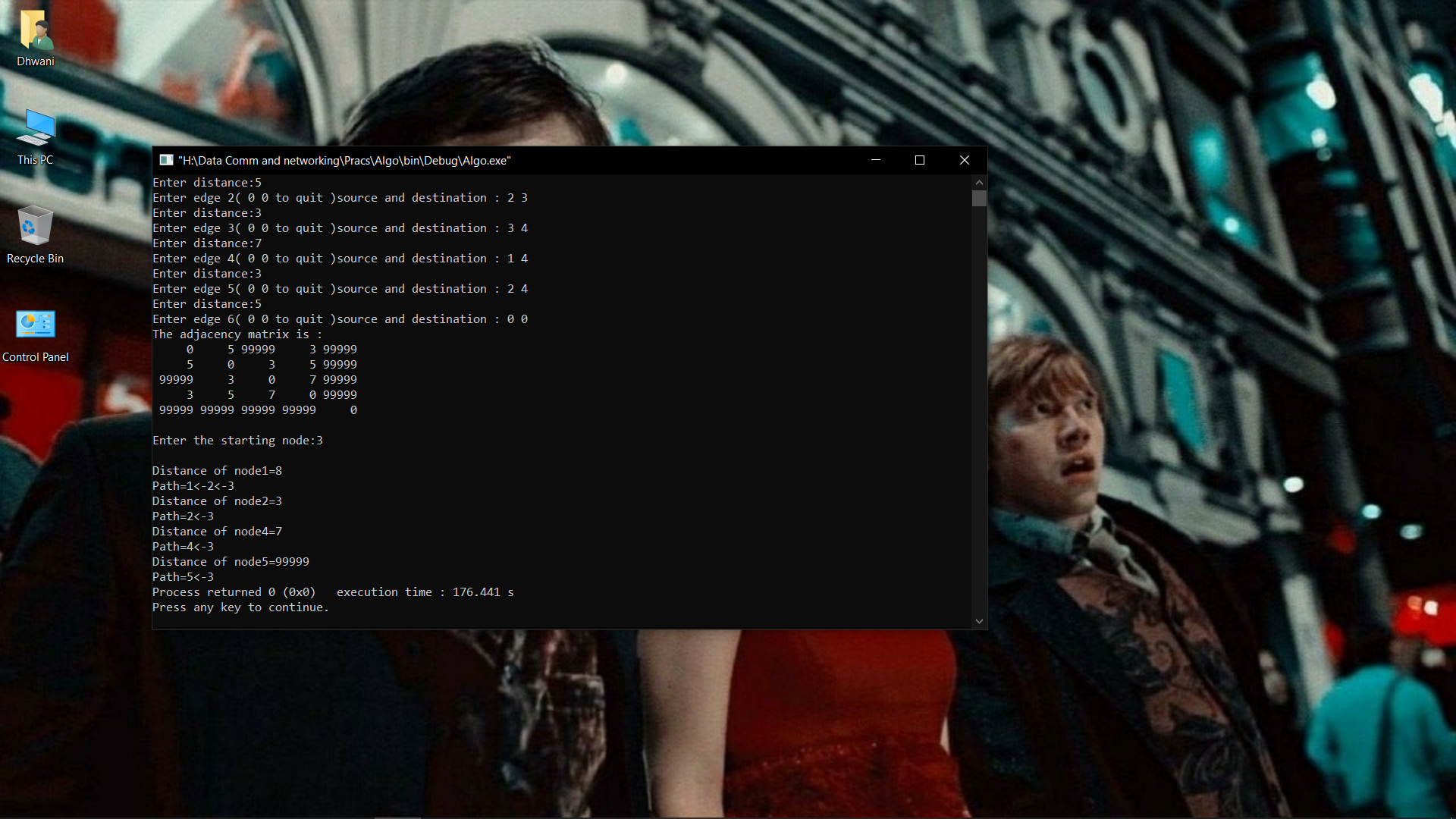
printf("<-%d",j);

}while(j!=startnode);

}

}

**Output:**



**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)Any or a combination of the above

**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** d) Its neighbors on

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

a) Static

b) Dynamic

c) Both

d) Any

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

a) Message

b) Frame

c) Datagram

d) None of the Above

**5) What is Traffic Shaping?**

**Ans.)**

 Traffic shaping, also  known as packet shaping, is a network management technique that delays certain types of packets to optimize overall network performance. For instance, an ISP may delay P2P packets, such as those transmitted by BitTorrent networks. The justification for throttling this type of traffic might be that it consumes a high amount of overall network bandwidth for the benefit of a small number of users. Applications such as BitTorrent clients may attempt to encrypt their data packets to make it difficult or impossible for an ISP to identify the traffic and prevent traffic shaping.

Traffic shaping, also known as "packet shaping," is the practice of regulating network data transfer to assure a certain level of performance, quality of service (QoS) or return on investment (ROI). The practice involves delaying the flow of packets that have been designated as less important or less desired than those of prioritized traffic streams. Regulating the flow of packets into a network is known as "bandwidth throttling." Regulation of the flow of packets out of a network is known as "rate limiting."

Traffic shaping is used for a number of purposes:

Time-sensitive data may be given priority over traffic that can be delayed briefly with little-to-no ill effect.

A large ISP (Internet service provider) may shape the traffic of an independent reseller.

In a corporate environment, business-related traffic may be given priority over other traffic.

An ISP may limit bandwidth consumption for certain applications to reduce costs and create the capacity to take on additional subscribers. This practice can effectively limit a subscriber's "unlimited connection" and is often imposed without notification.

Traffic shaping could be an integral component of the proposed two-tiered Internet, in which certain customers or services would get traffic priority for a premium charge.

**Output:**

**CO:** Build the skills of subnetting and routing mechanisms

**Conclusion:**

In this experiment we learnt implementation of the shortest path algorithm. Here we implemented dijkstra’s algorithm and as result we found the shortest path as well as the cost (here minimum distance).