

AIM: Algorithm to implement Dijkstra's algorithm to compute the Shortest path through a graph

DESCRIPTION:

- The **Shortest Path Routing Algorithm** is used in computer networks to determine the path between a source and destination that minimizes cost, such as distance, time, or bandwidth usage.
- It represents the network as a weighted graph, where routers are nodes and links are edges with associated costs.
- Algorithms like **Dijkstra's** and **Bellman-Ford** are commonly used to compute the shortest path efficiently.
- This ensures optimal routing of data packets, reduces network congestion, and improves overall performance.

PROGRAM:

```
#include<stdio.h>

void main()
{
    int path[5][5],i,j,min,a[5][5],p,st=1,ed=5,stp,edp,t[5],index;
    printf("enter the cost matrix\n");
    for(i=0;i<5;i++)
        for(j=0;j<5;j++)
            scanf("%d",&a[i][j]);
    printf("enter the paths\n");
    scanf("%d",&p); printf("enter possible paths\n");
    for(i=0;i<p;i++)
        for(j=0;j<5;j++)
            scanf("%d",&path[i][j]);
    for(i=0;i<p;i++)
    {
        t[i]=0;stp=st;
        for(j=0;j<5;j++)
        {
            edp=path[i][j+1];
            if(edp>0)
```

```

t[i]=t[i]+a[stp][edp];
if(edp==ed)
break;
else
stp=edp;
}
}

min=t[st];
index=st;
for(i=0;i<p;i++)
{
if(min>t[i])
{
min=t[i];
index=i;
}
}

printf("minimum cost %d",min);
printf("\n minimum cost path ");
for(i=0;i<5;i++)
{
printf("--> %d",path[index][i]);
if(path[index][i]==ed)
break;
}
}
}

```

OUTPUT

```
enter the cost matrix
0 1 4 2 0
1 0 3 7 0
4 3 0 5 0
2 7 5 0 9
0 0 0 9 0
enter the paths
3
enter possible paths
1 2 4 5 0
1 3 4 5 0
1 4 5 0 0
minimum cost 0
minimum cost path --> 1--> 4--> 5

==== Code Exited With Errors ===
```