DHRUV GARG 16 BCE 1190

Data Structures and Algorithms

LAB 7 – Djikstra’s Algorithm

DJIKSTRAS’S STRATEGY

Code for shortest path using Djikstra’s strategy

```
#include<stdio.h>

#define INFINITY 9999

#define MAX 10


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

{

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

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

  for(i=0; i<n; i++)

    for(j=0; j<n; j++)

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

        cost[i][j]=INFINITY;

      else

        cost[i][j]=G[i][j];

  for(i=0; i<n; i++)

  {

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

       pred[i]=startnode;

       visited[i]=0;

 }


  distance[startnode]=0;

  visited[startnode]=1;

  count=1;
```

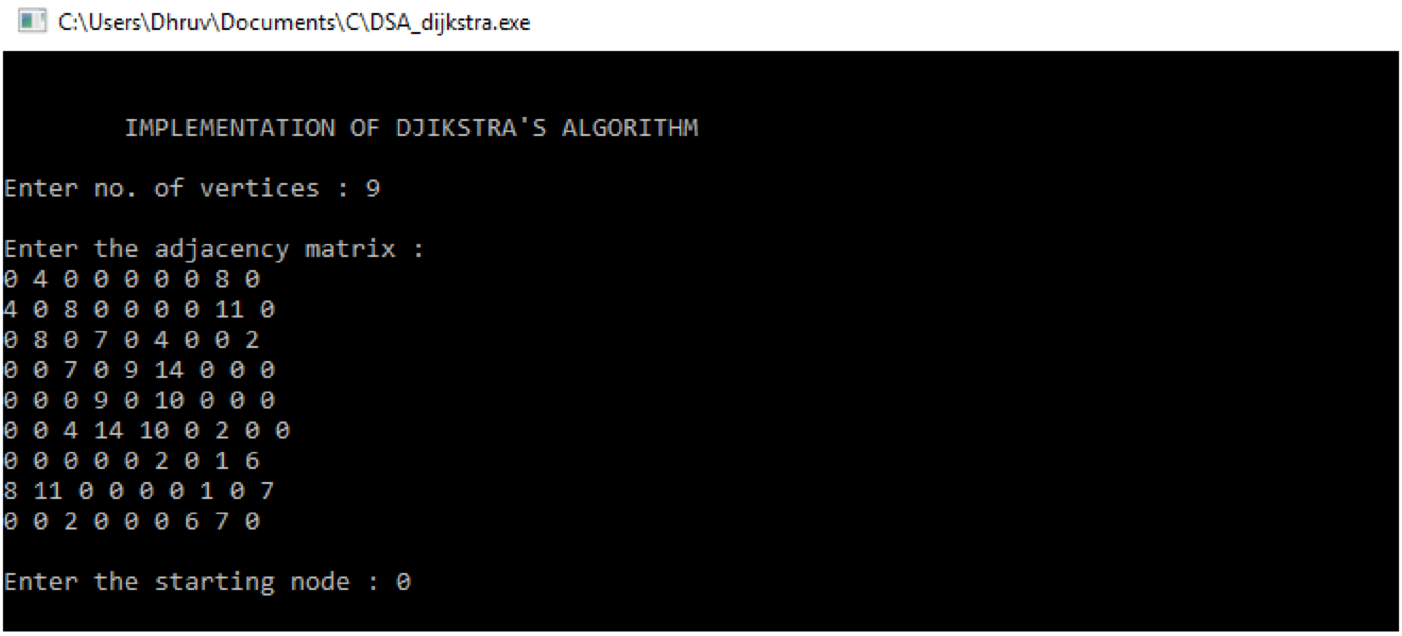
```
    while( count < n-1)
    {
        mindistance=INFINITY;
        for(i=0; i<n; i++)
            if(distance[i]<mindistance&&!visited[i])
            {
                mindistance=distance[i];
                nextnode=i;
            }
        visited[nextnode]=1;
        for(i=0; i<n; i++)
            if(!visited[i])
                if(mindistance+cost[nextnode][i]<distance[i])
                {
                    distance[i]=mindistance+cost[nextnode][i];
                    pred[i]=nextnode;
                }
        count++;
    }

    for(i=0; i<n; i++)
        if(i!=startnode)
        {
            printf("\n\nDistance of node %d = %d ",i,distance[i]);
            printf("\nPath = %d ",i);
            j=i;
            do
            {
                j=pred[j];
                printf(" <- %d",j);
                }
                while(j!=startnode);
        }
}
```

```
int main()
{
   printf("\n\n\tIMPLEMENTATION OF DJIKSTRA'S ALGORITHM\n\n");
   int G[MAX][MAX],i,j,n,u;
   printf("Enter no. of vertices : ");
   scanf("%d",&n);
   printf("\nEnter the adjacency matrix : \n");

   for(i=0; i<n; i++)
      for(j=0; j<n; j++)
         scanf("%d",&G[i][j]);
   printf("\nEnter the starting node : ");
   scanf("%d",&u);
   dijkstra(G,n,u);
   return 0;
}
```

Screenshot for shortest path using Djikstra’s method



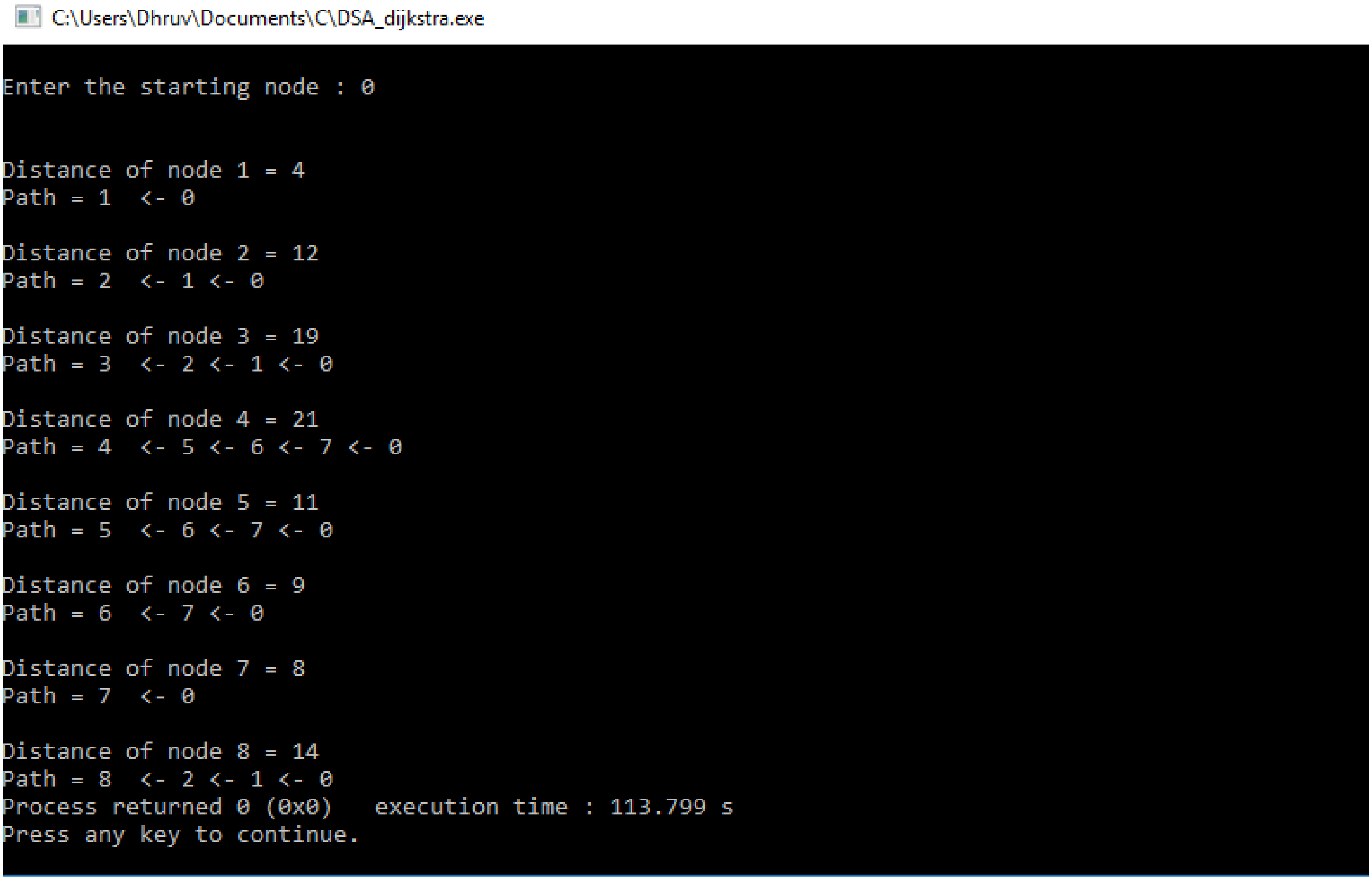
```


        IMPLEMENTATION OF DJIKSTRA'S ALGORITHM

Enter no. of vertices : 9

Enter the adjacency matrix :
0 4 0 0 0 0 0 8 0
4 0 8 0 0 0 0 11 0
0 8 0 7 0 4 0 0 2
0 0 7 0 9 14 0 0 0
0 0 0 9 0 10 0 0 0
0 0 4 14 10 0 2 0 0
0 0 0 0 0 2 0 1 6
8 11 0 0 0 0 1 0 7
0 0 2 0 0 0 6 7 0

Enter the starting node : 0
```



C:\Users\Dhruv\Documents\C\DSA_dijkstra.exe

```
Enter the starting node : 0


Distance of node 1 = 4
Path = 1  <- 0

Distance of node 2 = 12
Path = 2  <- 1 <- 0

Distance of node 3 = 19
Path = 3  <- 2 <- 1 <- 0

Distance of node 4 = 21
Path = 4  <- 5 <- 6 <- 7 <- 0

Distance of node 5 = 11
Path = 5  <- 6 <- 7 <- 0

Distance of node 6 = 9
Path = 6  <- 7 <- 0

Distance of node 7 = 8
Path = 7  <- 0

Distance of node 8 = 14
Path = 8  <- 2 <- 1 <- 0
Process returned 0 (0x0)   execution time : 113.799 s
Press any key to continue.
```