DAA Assignment 4

Name: Digvijay Pawar

Class: TY.Btech Comp B2

Gr.No: 21810344

Roll No: 322043

# ***Dynamic Programming Method (Multistage graph)***

***Code Implementation :***

#include <iostream>

#include <bits/stdc++.h>

#define MAX 20

using namespace std;

int stages,stage\_vertices[MAX],c[MAX][MAX]={9999};

int cost[MAX]={0},q[MAX],n;

int get\_min(int s,int n)

{

int min= INT\_MAX;//equal to infinity

int min\_vertex;

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

{

if(c[s][i]!=0)

{

if(min>(c[s][i]+cost[i]))

{

min=c[s][i]+cost[i];

min\_vertex=i;

}

}

}

return min\_vertex;

}

int main() {

int i,j,m,p,no\_of\_vertices=0;

cout<<"Enter no of vertices: "<<endl;

cin>>no\_of\_vertices;

cout<<"Enter no of stages : "<<endl;

cin>>stages;

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

{

cout<<"Enter no of vertices in stage: "<<i+1<<endl;

cin>>stage\_vertices[i];

}

i=0;

j=stage\_vertices[0];

for(m=0; m<stages; m++)

{

j=i+stage\_vertices[m];

for(; i<j; i++)

{

for(p=0; p<stage\_vertices[m+1]; p++)

{

cout<<"Enter cost for vertex:"<<i+1<<" to "<<p+j+1<<endl;

cin>>c[i][p+j];

}

}

}

n = no\_of\_vertices;

int x,r;

int d[20];

for(x=n-2; x>=0; x--)

{

r=get\_min(x,n);

cost[x]=c[x][r]+cost[r];

d[x]=r;

}

std::cout << "Minimum cost is :"<<cost[0] << '\n';

q[0]=0;

q[stages-1]=n-1;

for(i=1; i<stages-1; i++)

q[i]=d[q[i-1]];

int ind;

cout<<"Shortest path is: ";

for(ind=0; ind<stages-1; ind++)

cout<<q[ind]+1<<" ";

cout<<q[ind]+1<<endl;//printing target node

return 0;

}

***Output:***

