Md. Asad Chowdhury Dipu

Id: 2019-1-60-093

Section: 1

Ans#1(Topological Sort)

#include<stdio.h>

int main()

{

int i,j,k,n;

printf("Enter the no of vertex: ");

scanf("%d",&n);

int mat[20][20],InDegree[20],t[20];

int cnt=0;

printf("Enter the adjacency matrix:\n");

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

{

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

scanf("%d",&mat[i][j]);

}

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

{

InDegree[i]=0;

t[i]=0;

}

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

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

InDegree[i]=InDegree[i]+mat[j][i];

printf("\nThe Topological Order: ");

while(cnt<n)

{

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

{

if((InDegree[k]==0) && (t[k]==0))

{

printf("%d ",(k+1));

t[k]=1;

}

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

{

if(mat[i][k]==1)

InDegree[k]--;

}

}

cnt++;

}

return 0;

}

Ans#2(Ford-Fulkerson Algorithm for Maximum Flow Problem)

/\*Ford-Fulkerson Algorithm for Maximum Flow Problem\*/

#include<stdio.h>

#include<bits/stdc++.h>

using namespace std;

bool bfs(int ResGraph[][6],int s,int t,int parent[])

{

bool visit[6];

memset(visit,0,sizeof(visit));

queue <int> Q;

Q.push(s);

visit[s]=true;

parent[s]=-1;

while(!Q.empty())

{

int i=Q.front();

Q.pop();

for (int j=0; j<6; j++)

{

if(visit[j]==false && ResGraph[i][j]>0)

{

Q.push(j);

parent[j]=i;

visit[j]=true;

}

}

}

return (visit[t]==true);

}

void Ford\_Fulkerson(int graph[100][100],int s,int t)

{

int i,j;

int ResGraph[6][6];

for (i=0; i<t+1; i++)

{

for (j=0; j<t+1; j++)

{

ResGraph[i][j]=graph[i][j];

}

}

int p[t+1];

int MaxFlow=0;

while(bfs(ResGraph,s,t,p))

{

int PathFlow=INT\_MAX;

for(j=t; j!=s; j=p[j])

{

i=p[j];

PathFlow=min(PathFlow, ResGraph[i][j]);

}

for(j=t; j!=s; j=p[j])

{

i=p[j];

ResGraph[i][j]-=PathFlow;

ResGraph[j][i]+=PathFlow;

}

MaxFlow+=PathFlow;

}

printf("\nMax flow : %d\n\n",MaxFlow);

}

int main()

{

int graph[100][100];

int n,e,u,v,w,i,j;

printf("Enter number of vertex: ");

scanf("%d",&n);

printf("Enter number of edge: ");

scanf("%d",&e);

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

{

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

{

graph[i][j]=0;

}

}

printf("Enter edges with capacity:\n");

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

{

scanf("%d%d%d",&u,&v,&w);

graph[u][v]=w;

}

printf("\n");

Ford\_Fulkerson(graph,0,n);

return 0;

}