1. DFS **(17/Apr/2020)**
   1. Store your graph using adjacency matrix(as we have done in the lab session)
   2. Write separate functions for the important operations(so that u can reuse those functions in the next programs too)
   3. DFS() should visit the nodes in the correct order and print them based on the increasing order of their Finishing time

Source Code :

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

int mat[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];

int main()

{

int m;

cout <<"Enter no of vertices:";

cin >> n;

cout <<"Enter no of edges:";

cin >> m;

for(k=1; k<=m; k++)

{

cin >>i>>j;

mat[i][j]=1;

}

cout<<" \n Adjacency Matrix : \n";

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

{

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

{

cout<<mat[i][j]<<"\t";

}

cout<<endl;

}

cout <<"Enter initial vertex to traverse from:";

cin >>v;

cout <<"DFS ORDER:";

cout << v <<" ";

visited[v]=1;

k=1;

while(k<n)

{

for(j=n; j>=1; j--)

if(mat[v][j]!=0 && visited[j]!=1 && visit[j]!=1)

{

visit[j]=1;

stk[top]=j;

top++;

}

v=stk[--top];

cout<<v << " ";

k++;

visit[v]=0;

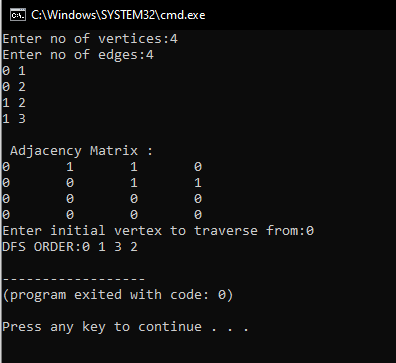
visited[v]=1;

}

return 0;

}

**OUTPUT**



[**https://github.com/harinarayanank/Competitive-Lab**](https://github.com/harinarayanank/Competitive-Lab/tree/master/Sorting%20Algorithms%20Set%201)