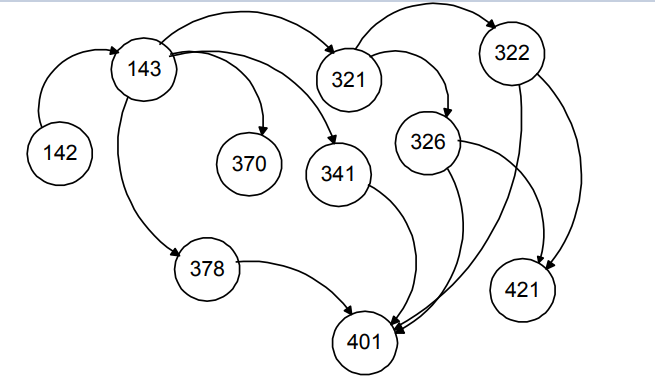
Lab Assignment 5 (Graded, max marks =15, Submission deadline: November 08, 2018 (1700hrs))

Graph Theory & Algorithms

(Fall 2018)

Name: **ABDULLAH AFZAL RAJA**

The directed graph shown below represents dependencies of different courses in a university.



Perform the following tasks on the graph

1. Traverse the graph using DFS and show the output sequence
2. Sort the graph in a Topological order to find out at-least one order in which these courses can be taken
3. Write python code using Networkx library to perform task (i) and (ii)

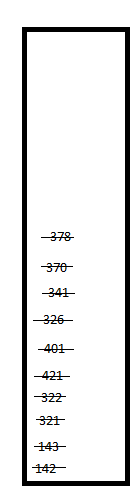
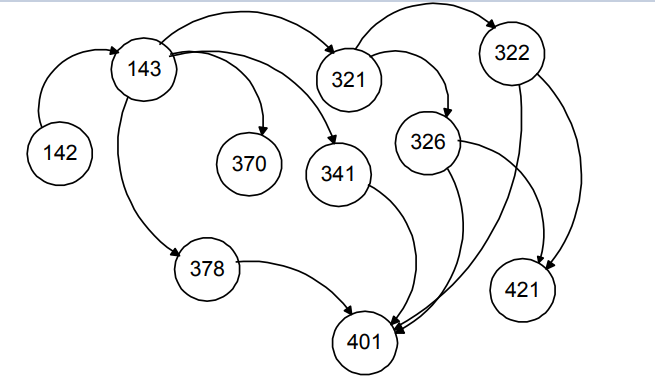
**Instructions:** In order to solve this assignment, you can use books, notes, slides etc. You can also collaborate with any students within your class. However, you must specify and acknowledge them by providing the following information.

Collaborator 1:Fakhar Abbas

Collaborator 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Answer:

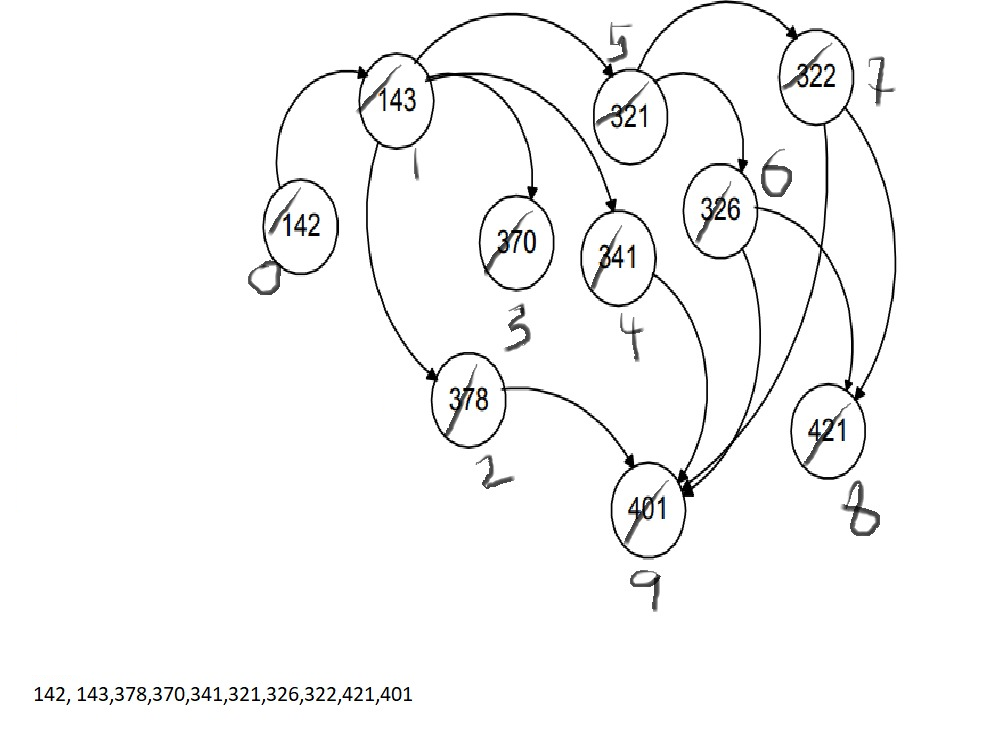
Stack:



Output:

Output:142, 143, 321, 322, 421, 401, 326, 341, 370, 378

(ii) Answer:



As there can be several orders in which one can take all of these courses, The only thing remains the same is the dependencies of the node, e.g. course “143” cannot be taken before course “142”.

Following is the order we got when we applied DFS and chose ascending order .

142-143-378-370-341-321-326-322-421-401

(iii) **Python** **Code:**

import networkx as nx

import matplotlib.pyplot as plt

G = nx.DiGraph()

G.add\_node("142")

G.add\_node("143")

G.add\_node("378")

G.add\_node("370")

G.add\_node("341")

G.add\_node("321")

G.add\_node("326")

G.add\_node("322")

G.add\_node("421")

G.add\_node("401")

G.add\_edge("142" , "143")

G.add\_edge("143" , "321")

G.add\_edge("143" , "341")

G.add\_edge("143" , "370")

G.add\_edge("143" , "378")

G.add\_edge("321" , "322")

G.add\_edge("321" , "326")

G.add\_edge("322" , "421")

G.add\_edge("322" , "401")

G.add\_edge("326" , "421")

G.add\_edge("326" , "401")

G.add\_edge("341" , "401")

G.add\_edge("378" , "401")

nx.draw(G, with\_labels =True)

plt.show()

#print(nx.is\_connected(G))

print("DFS Transversal")

dfs = list(nx.dfs\_edges(G))

print(dfs)

print("Topological Order")

topolological = list(nx.topological\_sort(G))

print(topolological)