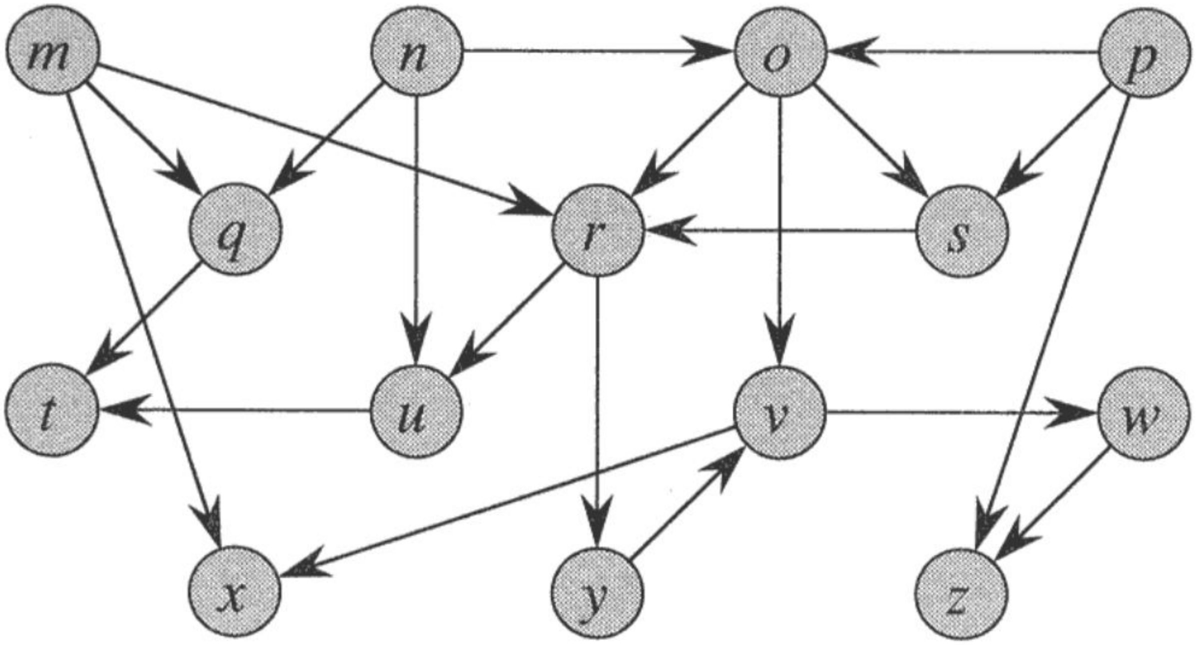
Problem 1. - 15 points

Perform DFS on the Directed Acyclic Graph (DAG) shown on the last slide of "Cormen-chapter22.pptx" file and draw the final topological sort graph. Scanned/snapshot of hand solution paper will work.



HW3 Problem 1

-1	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø
m	n	o	p	q	r	s	t	u	v	w	x	y	z

x
m

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	13	-1	-1
m	n	o	p	q	r	s	t	u	v	w	x	y	z

t
q
m

-1	-1	-1	-1	-1	-1	-1	12	-1	-1	-1	13	-1	-1
m	n	o	p	q	r	s	t	u	v	w	x	y	z

q
m

-1	-1	-1		11			12				13		
m	n	o	p	q	r	s	t	u	v	w	x	y	z

u
r
M

				11			12	10			13		
m	n	o	p	q	r	s	t	u	v	w	x	y	z

z
w
v
y
r
m

				11			12	10			13		9
m	n	o	p	q	r	s	t	u	v	w	x	y	Z

w
v
y
r
M

				11			12	10		8	13		9
m	n	o	p	q	r	s	t	u	v	w	x	y	z

v
y
r
m

				11			12	10	7	8	13		9
m	n	o	p	q	r	s	t	u	v	w	x	y	z

y
r
m

				11			12	10	7	8	13	6	9
m	n	o	p	q	r	s	t	u	v	w	x	y	z

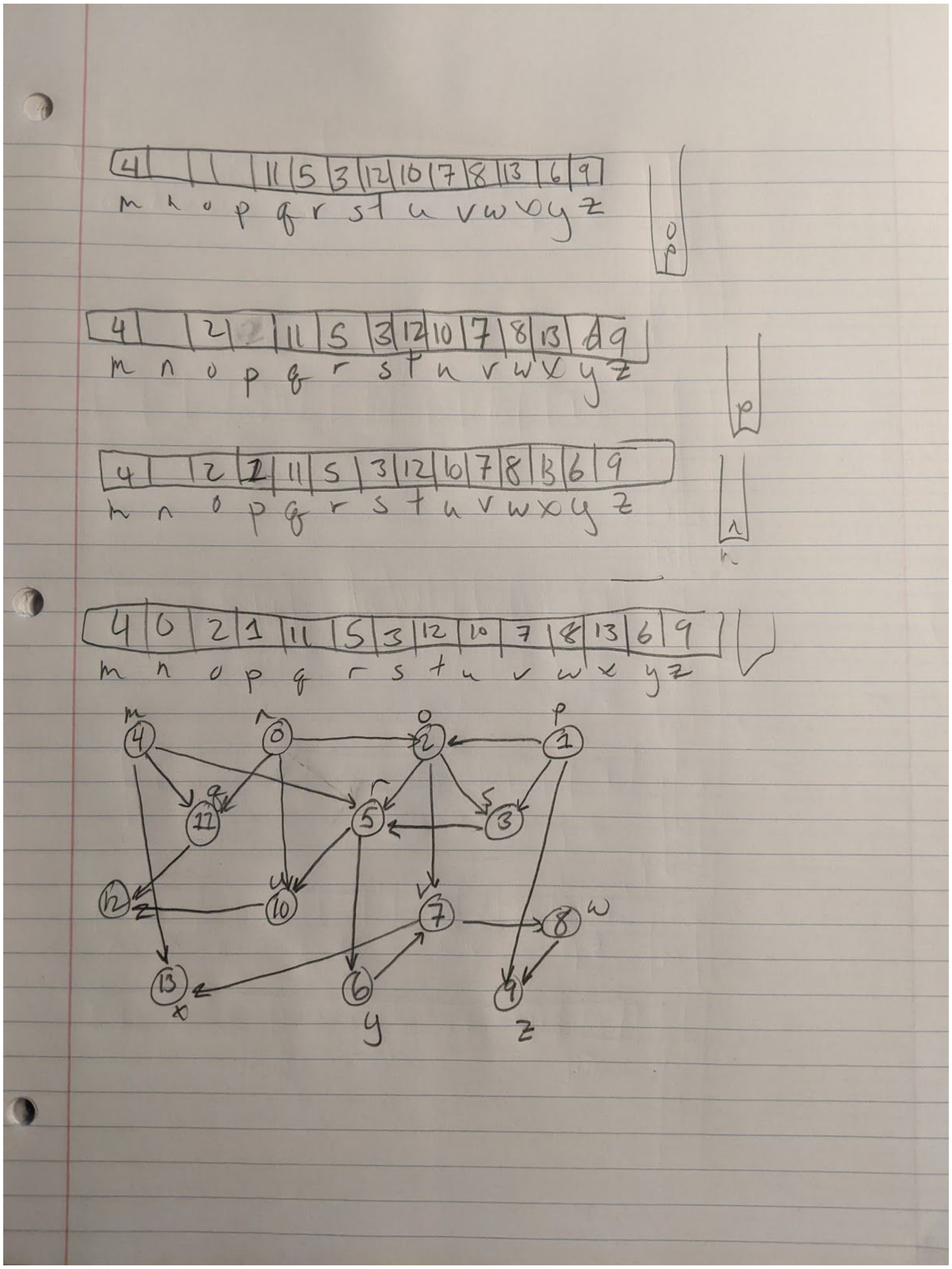
r
m

				11	5		12	10	7	8	13	6	9
m	n	o	p	q	r	s	t	u	v	w	x	y	z

m

4				11	5		12	10	7	8	13	6	9
m	n	o	p	q	r	s	t	u	v	w	x	y	z

s
o
p



Problem 2. - 15 points

Develop a C++/Java/Python program to find the minimum spanning tree (MST) using Prim's Algorithm. Consider file I/O. The 2D adjacency matrix (where nonzero values are the weights of the corresponding edges) will be taken from a file. In each row, the values are space separated. The output will be shown on the screen as an adjacency matrix.

The maximum number of nodes to be handled is 10.

Sample Input (for a graph with 6 nodes). Inputs will be taken from a file:

0 4 0 0 0 8

4 0 7 0 0 11

0 7 0 2 3 0

0 0 2 0 6 7

0 0 3 6 0 1

8 11 0 7 1 0

For better understanding of the input format, here attached the diagram of the sample graph representing the sample input.

HW3 MST.jpg

Sample Output (will be displayed on screen):

0 4 0 0 0 0

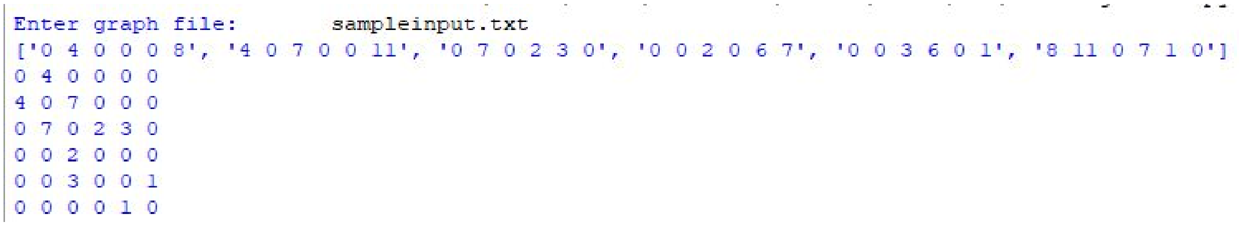
4 0 7 0 0 0

0 7 0 2 3 0

0 0 2 0 0 0

0 0 3 0 0 1

0 0 0 0 1 0



```
Enter graph file:       sampleinput.txt
['0 4 0 0 0 8', '4 0 7 0 0 11', '0 7 0 2 3 0', '0 0 2 0 6 7', '0 0 3 6 0 1', '8 11 0 7 1 0']
0 4 0 0 0 0
4 0 7 0 0 0
0 7 0 2 3 0
0 0 2 0 0 0
0 0 3 0 0 1
0 0 0 0 1 0
```

Problem 3. - 15 points

Write a C++/Java/Python program to find the shortest path from a single source to all other nodes in a directed weighted graph using **Dijkstra's Algorithm** (no negative weight). The 2D adjacency matrix (where nonzero values are the weights of the corresponding edges) will be taken from a file. In each row, the values are space separated. In the input matrix, the direction is denoted as Node-in-Row --> Node-in-Column. For example, if the [2, 3] value is 5, it means there is a directed edge from node2 -> node3 and the weight is 5.

The maximum number of nodes to be handled is 9.

Sample Input for a graph with 5 nodes (will be taken from file):

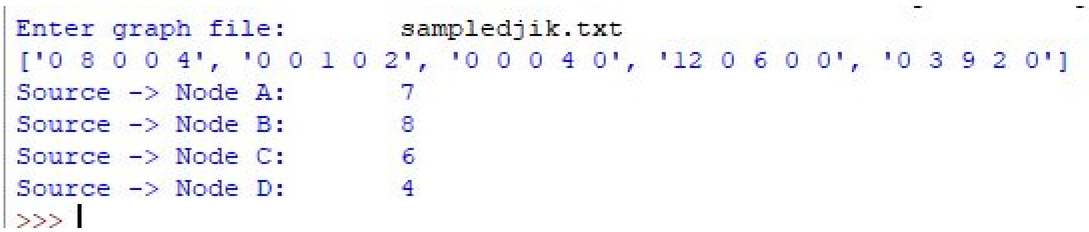
0 8 0 0 4

0 0 1 0 2

0 0 0 4 0

12 0 6 0 0

0 3 9 2 0



```
Enter graph file:       sampledjik.txt
['0 8 0 0 4', '0 0 1 0 2', '0 0 0 4 0', '12 0 6 0 0', '0 3 9 2 0']
Source -> Node A:       7
Source -> Node B:       8
Source -> Node C:       6
Source -> Node D:       4
>>> 
```