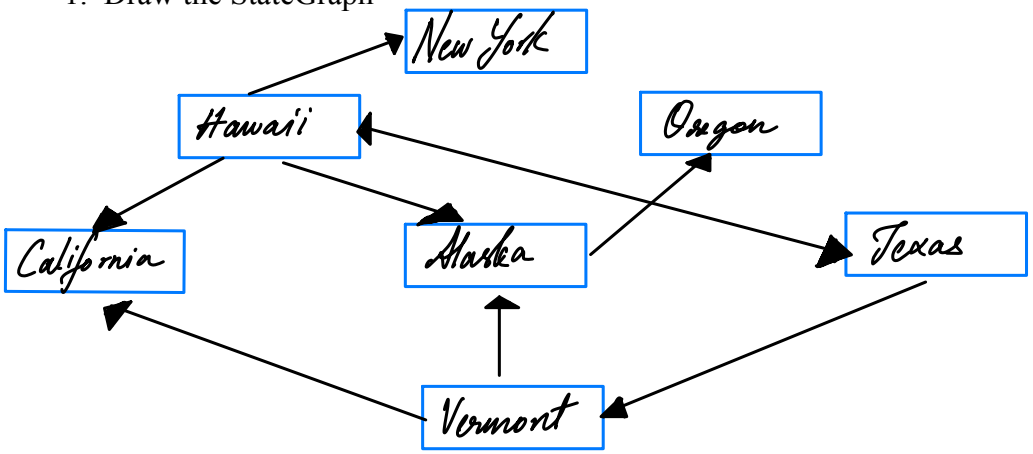
Nikitta Joan

CMSC204
Kartchner

V(StateGraph) = {Oregon, Alaska, Texas, Hawaii, Vermont, NewYork, California}
E(StateGraph) = {(Alaska, Oregon), (Hawaii, Alaska), (Hawaii, Texas), (Texas, Hawaii), (Hawaii, California), (Hawaii, New York), (Texas, Vermont), (Vermont, California), (Vermont, Alaska)}

1. Draw the StateGraph



1. Describe the graph pictured above, using the formal graph notation.

V(StateGraph) = {Oregon, Alaska, Texas, Hawaii, Vermont, New York, California}

E(StateGraph) = {(Alaska, Oregon), (Hawaii, Alaska), (Hawaii, Texas), (Texas, Hawaii), (Hawaii, California), (Hawaii, New York), (Texas, Vermont), (Vermont, California), (Vermont, Alaska)} -1

2. a. Is there a path from Oregon to any other state in the graph? No

   b. Is there a path from Hawaii to every other state in the graph? Yes

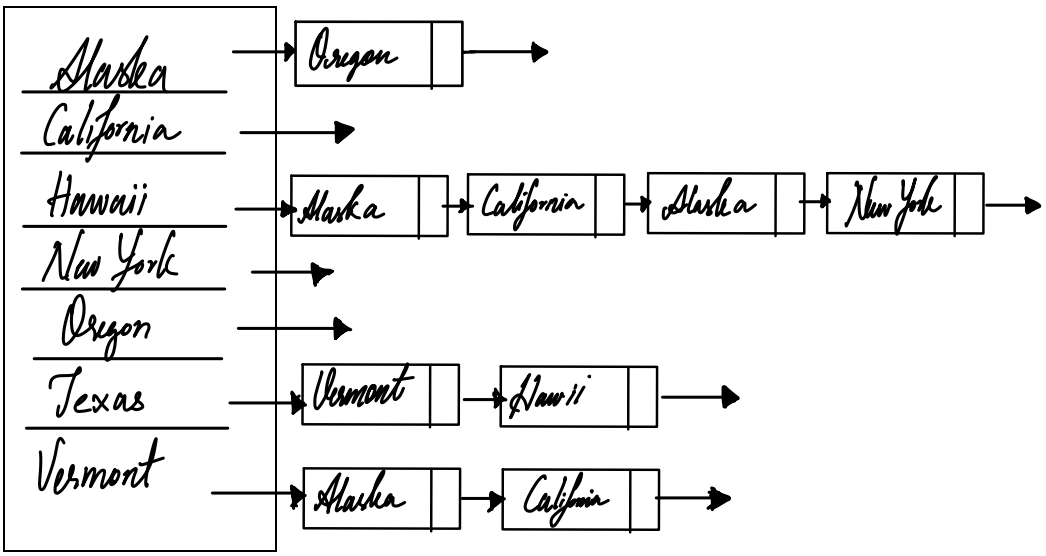
   c. From which state(s) in the graph is there a path to Hawaii? Texas

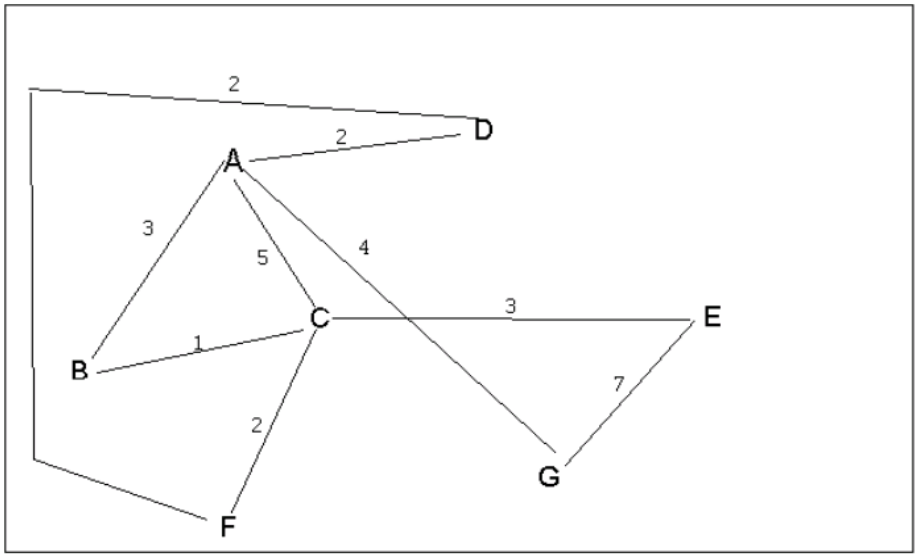
3. a. Show the adjacency matrix that would describe the edges in the graph.
Store the vertices in alphabetical order

States

Alaska	0	0	0	0	1	0	0	
California	0	0	0	0	0	0	0	
Hawaii	1	1	0	1	0	1	0	
New York	0	0	0	0	0	0	0	
Oregon	0	0	0	0	0	0	0	
Texas	0	0	1	0	0	0	1	
Vermont	1	1	0	0	0	0	0	

3. b. Show the adjacency lists
that would describe the edges in the graph





4 a. Which of the following lists the graph nodes in depth first order beginning with E?

A) E, G, F, C, D, B, A

B) G, A, E, C, B, F, D

C) E, G, A, D, F, C, B

D) E, C, F, B, A, D, G

option C

4 b. Which of the following lists the graph nodes in breadth first order beginning at F?

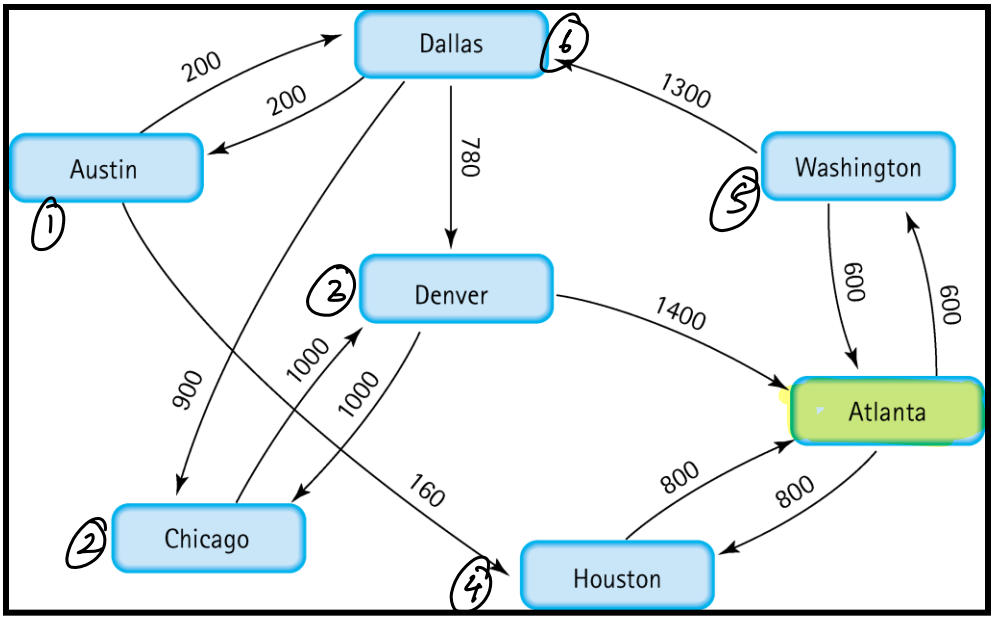
A) F, C, D, A, B, E, G

B) F, D, C, A, B, C, G

C) F, C, D, B, G, A, E

D) a, b, and c are all breadth first traversals

option A



5. Find the shortest distance from Atlanta to every other city

(1) Atlanta – Austin 600 + 1300 + 200 = 2100

(2) Atlanta – Chicago 600 + 1300 + 900 = 2800

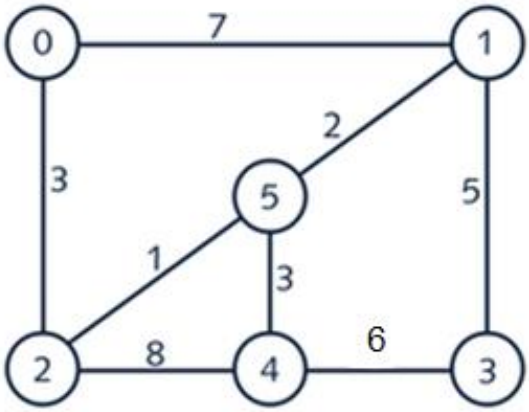
(3) Atlanta – Denver 600 + 1300 + 780 = 2680

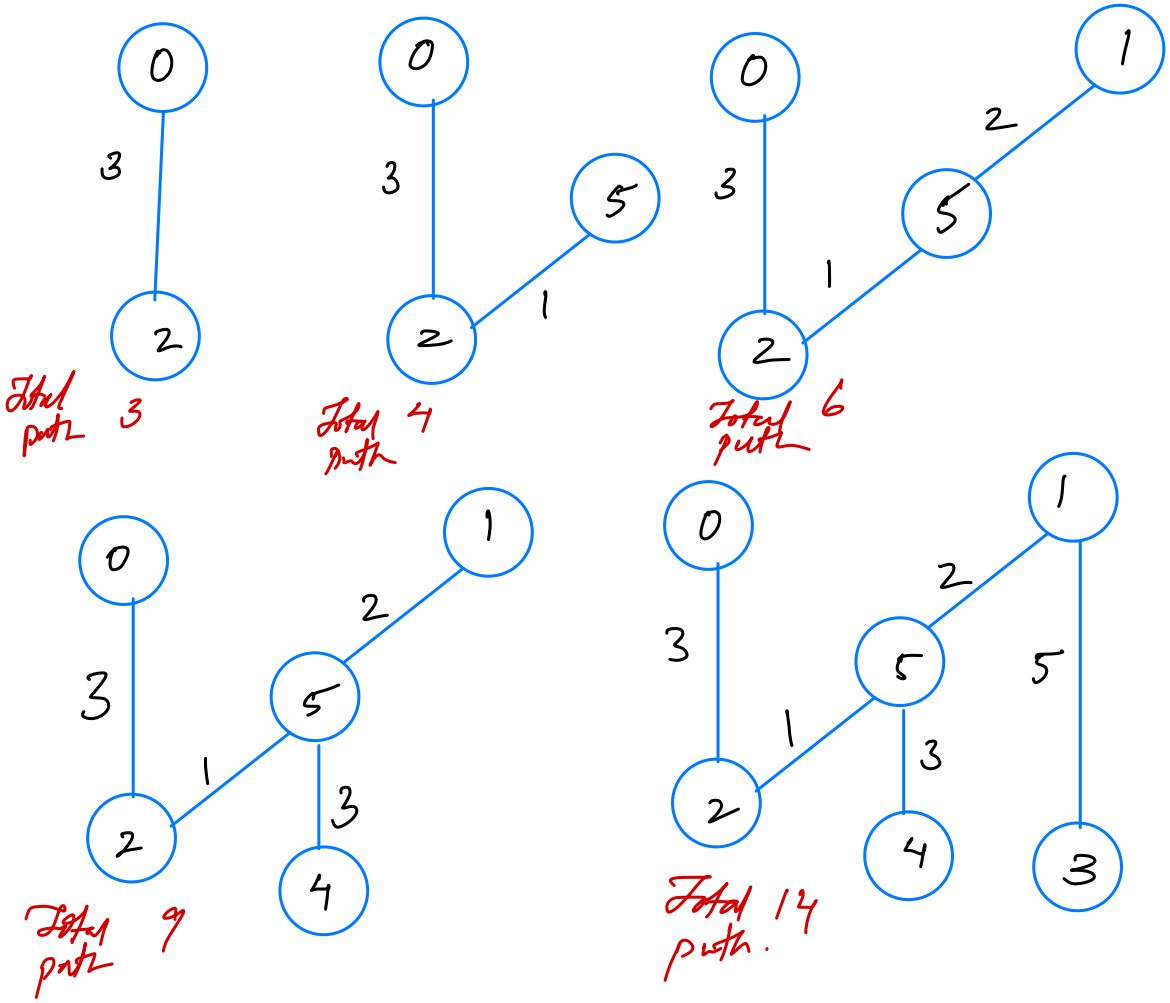
(4) Atlanta – Houston 800

(5) Atlanta – Washington 600

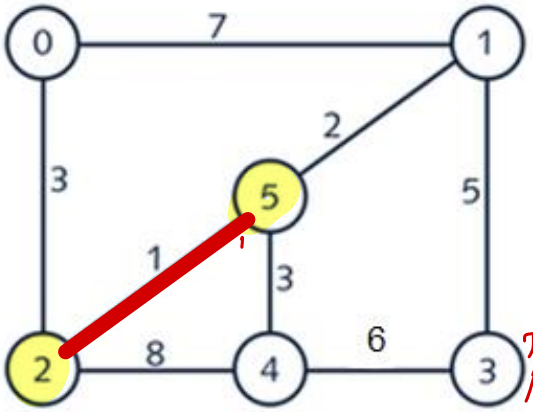
(6) Atlanta – Dallas 600 + 1300 = 1900

6. Find the minimal spanning tree using Prim's algorithm. Use 0 as the source vertex . Show the steps.

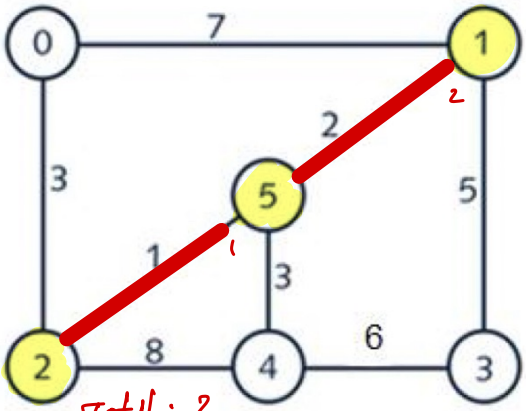


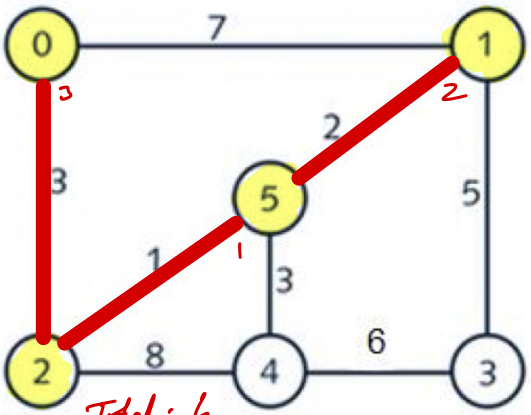


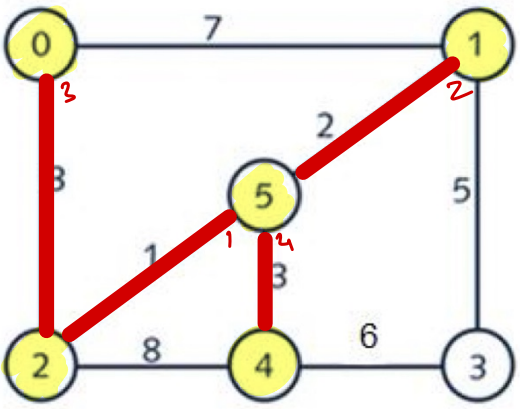
7. Find the minimal spanning tree using Kruskal's algorithm. Show the weights in order and the steps.

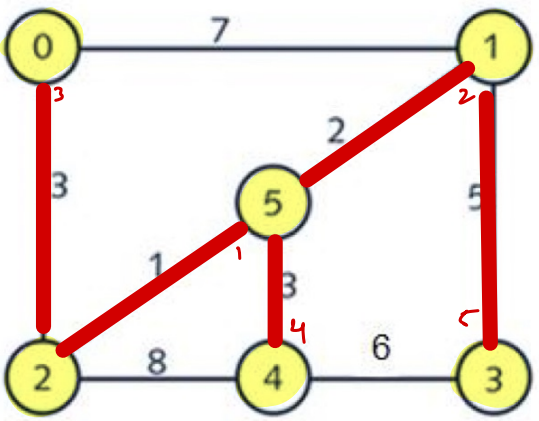


2-5 : 1
1-5 : 2
0-2 : 3
4-5 : 3
1-3 : 5
3-4 : 6
0-1 : 7
2-4 : 8

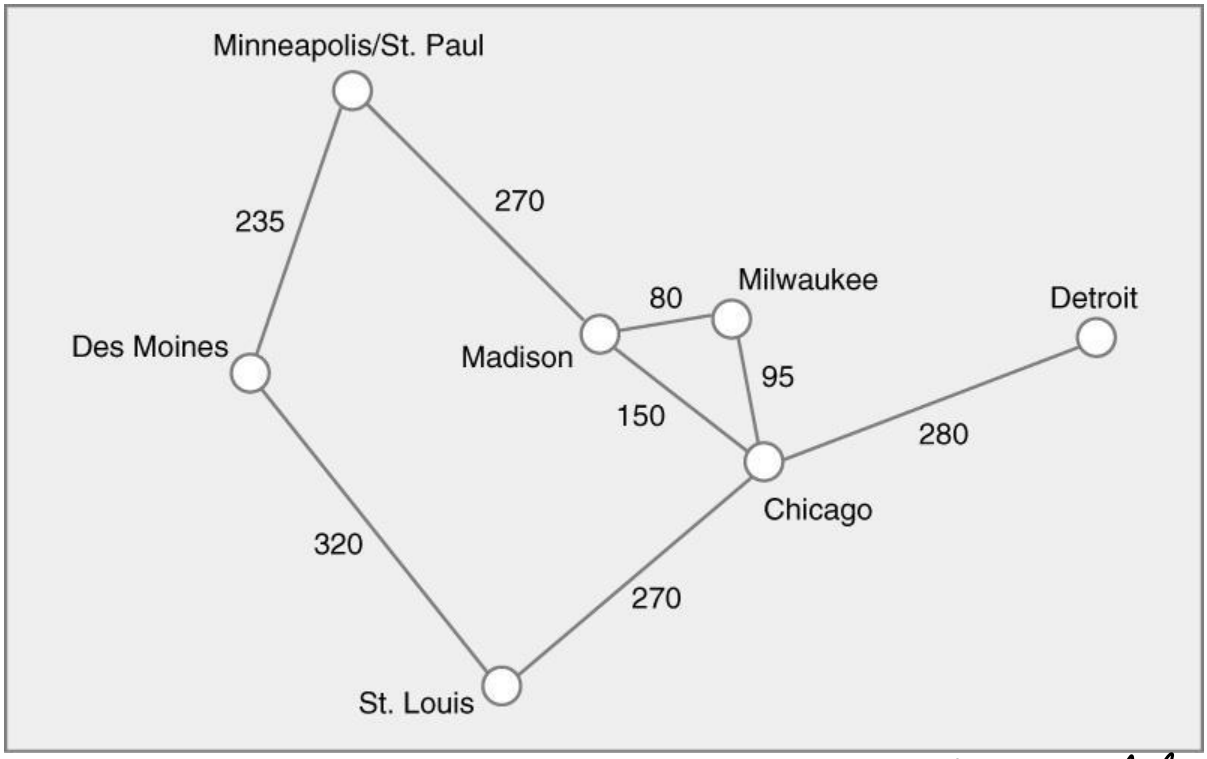








8. Find the minimal spanning tree using the algorithm you prefer. Use Minneapolis/St. Paul as the source vertex



Using Kruskal's

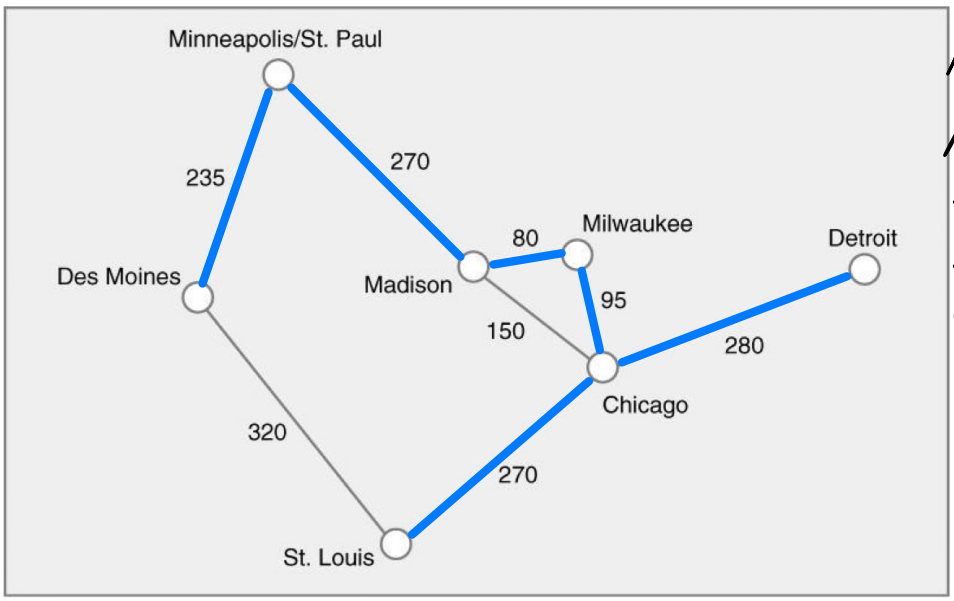
Madison – Milwaukee 80

Milwaukee – Chicago 95

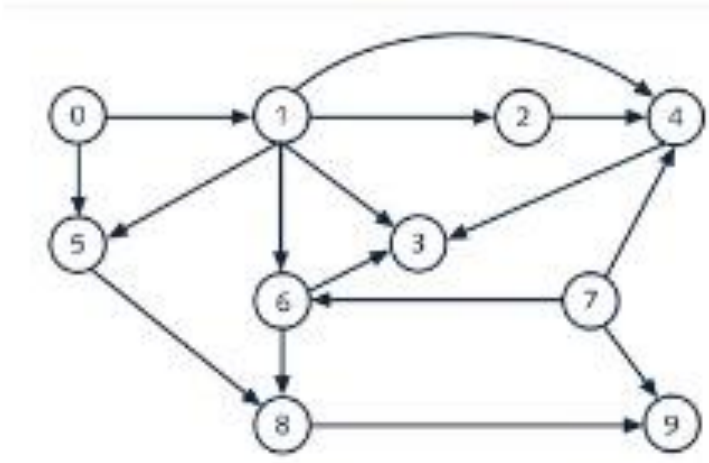
Minneapolis – Des Moines 235

Minneapolis – Madison 270

Chicago – St Louis 270



9. List the nodes of the graph in a breadth first topological ordering. Show the steps using arrays predCount, topologicalOrder and a queue



PredCount

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	1	1	3	3	2	2	0	2	2

TopologicalOrder

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

queue

0, 7

PredCount

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	1	3	3	1	2	0	2	2

TopologicalOrder

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0									

queue

7, 1

Pred Count

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	1	3	2	1	1	0	2	1

Topological Order

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	7								

queue

1

Pred Count

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	0	2	1	0	0	0	2	1

Topological Order

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	7	1							

queue

2, 5, 6

Pred Count

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	0	1	0	0	0	0	0	1

Topological Order

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	7	1	2	5	6				

queue

4, 8

Pred Count

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	0	0	0	0	0	0	0	0

Topological Order

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	7	1	2	5	6	4	8		

queue

3, 9

Pred Count

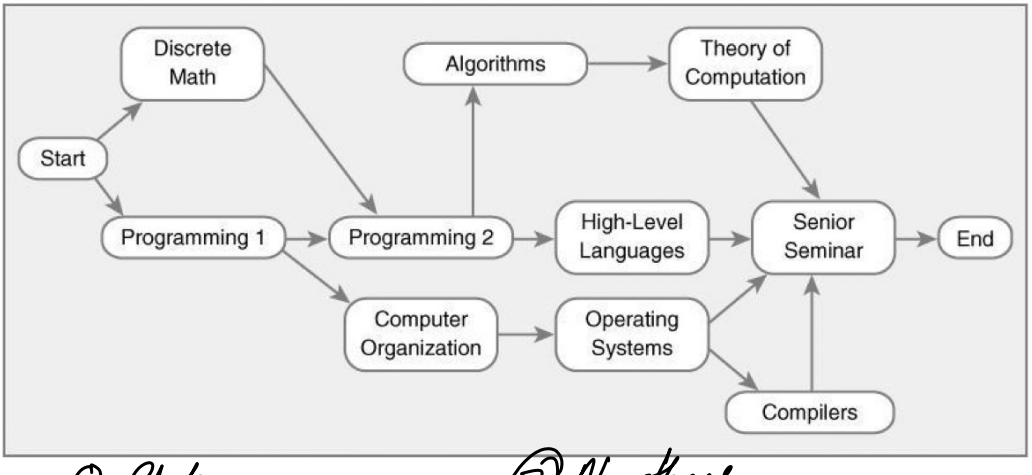
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	0	0	0	0	0	0	0	0	0

Topological Order

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	7	1	2	5	6	4	8	3	9

queue

10. List the nodes of the graph in a breadth first topological ordering.



1. Start
2. Discrete Math
3. Programming 1
4. Computer Organization
5. Programming 2
6. Operating Systems
7. Algorithms
8. High level languages
9. Compilers
10. Theory of Computation
11. Senior Seminar
12. End.