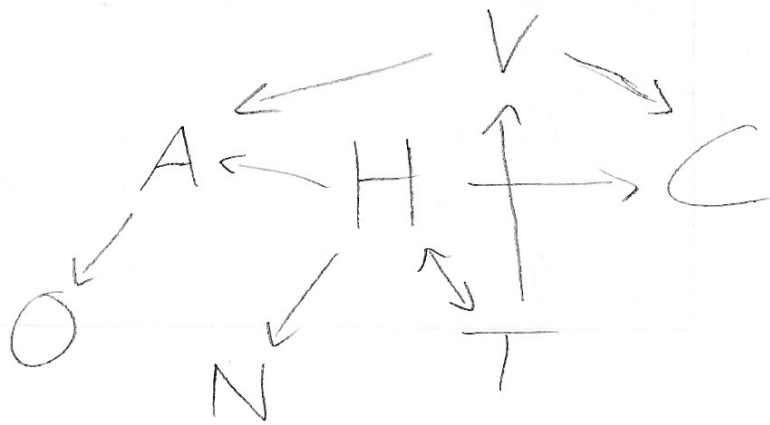
Ben Wiersma

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) = {Alaska, California, Hawaii, NewYork, Oregon, Texas, Vermont}

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

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?

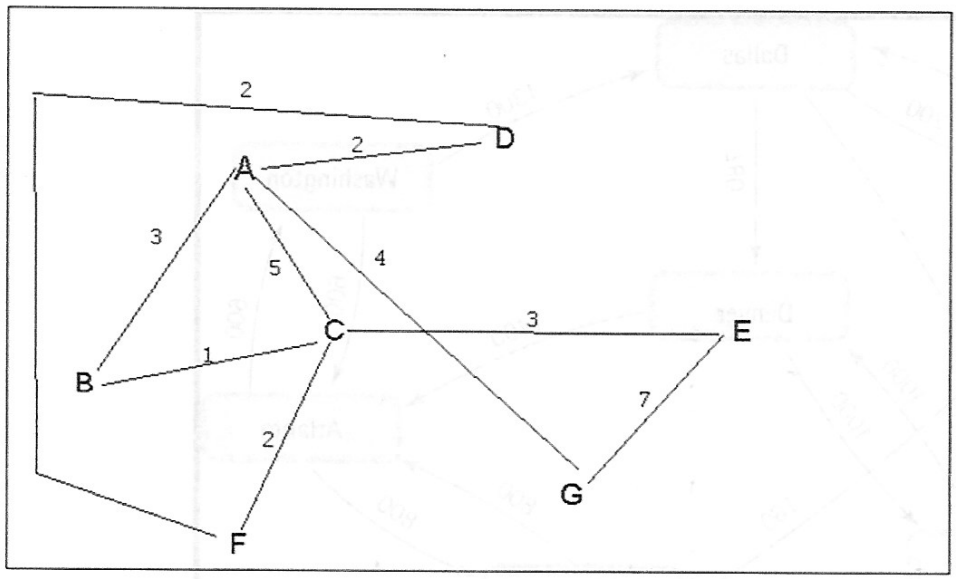
   Texas

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

States		A	C	H	N	O	T	V
Alaska	A	0	0	0	0	1	0	0
California	C	0	0	0	0	0	0	0
Hawaii	H	1	1	0	1	0	1	0
NewYork	N	0	0	0	0	0	0	0
Oregon	O	0	0	0	0	0	0	0
Texas	T	0	0	1	0	0	0	1
Vermont	V	1	1	0	0	0	0	0

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

Alaska	→ Oregon
California	
Hawaii	→ Alaska | California | NewYork | Texas
NewYork	
Oregon	
Texas	→ Hawaii | Vermont
Vermont	→ Alaska California



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

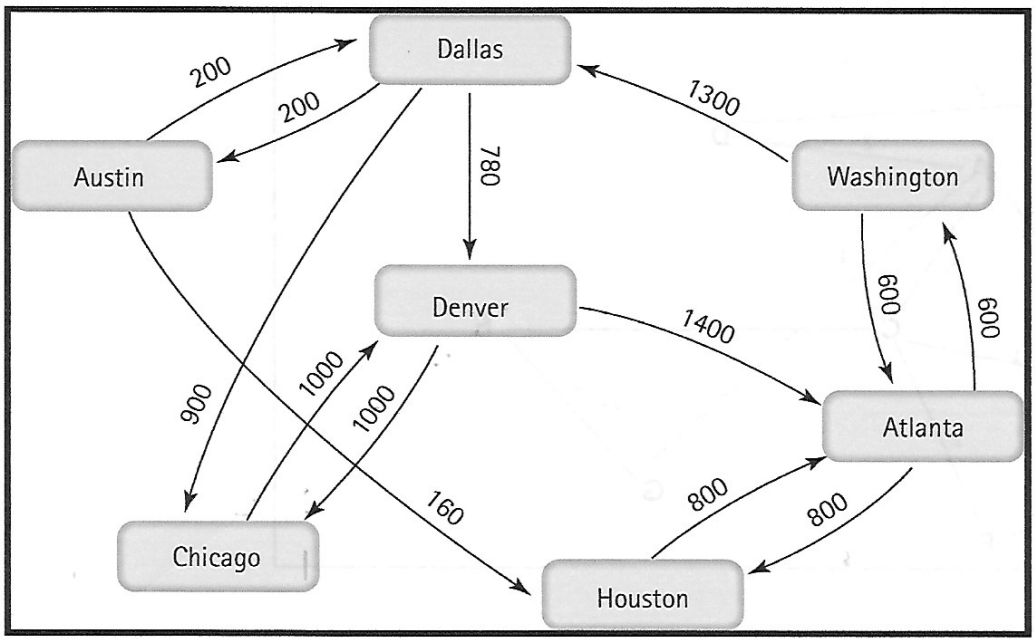
4 b. Which of the following lists the graph nodes in breadth first order beginning at F?

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



5. Find the shortest <u>distance</u> from Atlanta to every other city

Houston: 800

Washington: 600

Denver: 1,680

Washington → Dallas → Denver

Dallas: 1,900

Washington → Dallas

Chicago: 2,800

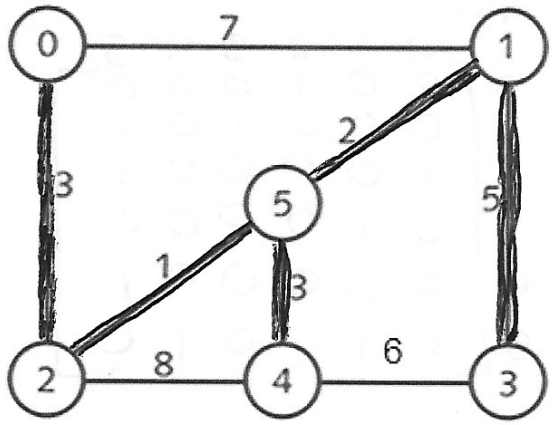
Washington → Dallas → Chicago

Austin: 2,100

Washington → Dallas → Austin

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

MST = {0, 2, 5, 1, 4, 3}



	0	1	2	3	4	5
0	0	0	1	0	0	0
1	0	0	0	1	0	1
2	1	0	0	0	0	1
3	0	1	0	0	0	0
4	0	0	0	0	0	1
5	0	1	1	0	1	0

1) Place 0 as source vertex into MST

2) Consider edges not in tree

0 -> 1(7), 2(3)

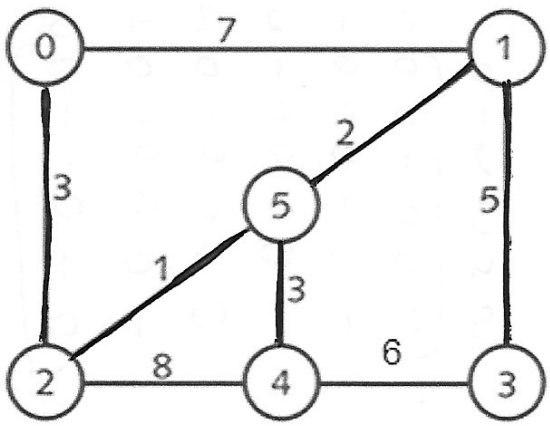
0, 2 -> 1(7), 5(1), 4(8)

0, 2, 5 -> 1(7), 4(8), 1(2)[through 5], 4(3)[through 5]

0, 2, 5, 1 -> 4(3[through 5], 8[through 2]), 3(5[through 1], 6[through 4])

0, 2, 5, 1, 4 -> 3(5[through 1], 6[through 4])

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



MST

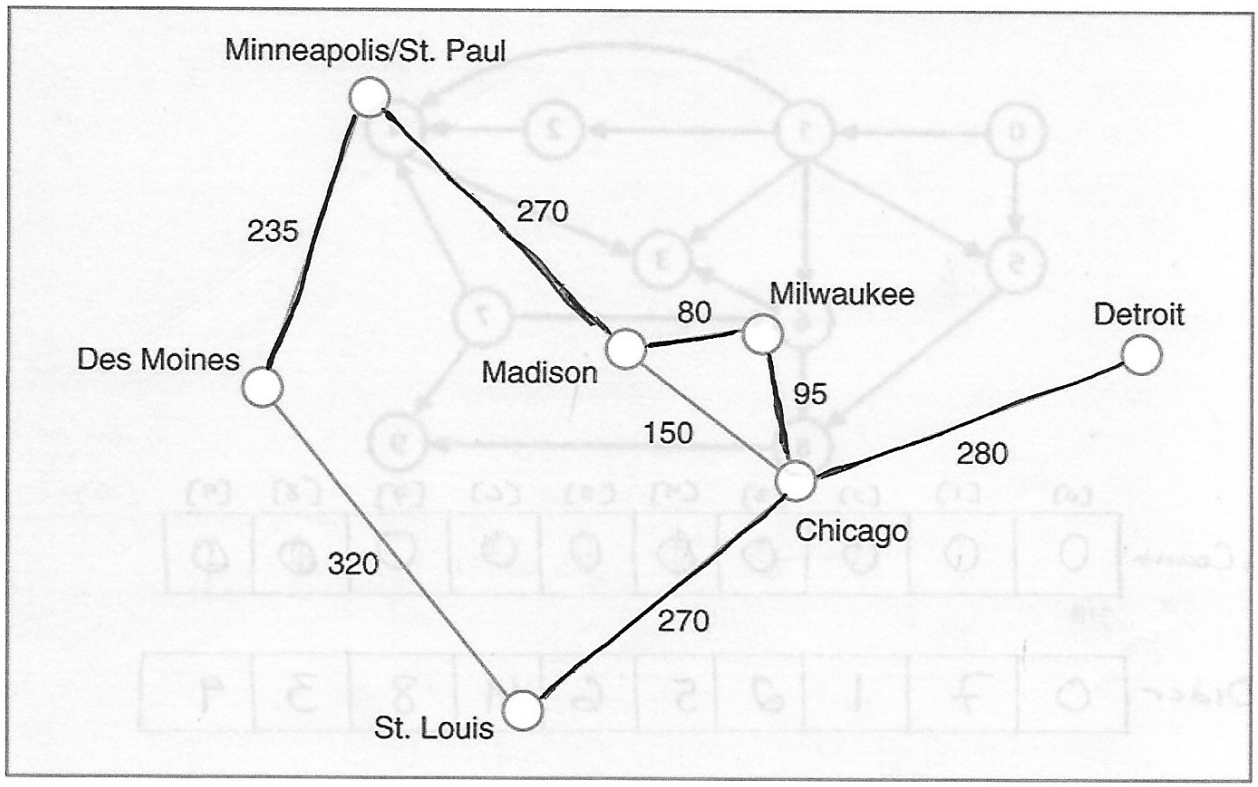
	0	1	2	3	4	5
0	0	0	1	0	0	0
1	0	0	0	1	0	1
2	1	0	0	0	0	1
3	0	1	0	0	0	0
4	0	0	0	0	0	1
5	0	1	1	0	1	0

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

→

[1] 2-5 ✓
[2] 1-5 ✓
[3] 0-2 ✓
[3] 4-5 ✓
[5] 1-3 ✓
[6] 3-4 X
[7] 0-1 X
[8] 2-4 X

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



[235] Minn/St.P - D.M.
[320] D.M. - St.L
[270] St.L - Chi
[280] Chi - Det
[95] Chi - Milw
[150] Chi - Mad
[80] Mad - Milw
[270] Mad - Minn/St.P

→

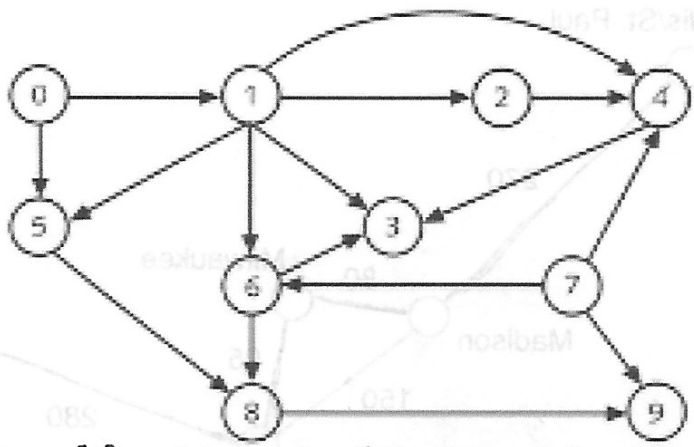
[80] Mad - Milw
[95] Chi - Milw
[150] Chi - Mad
[235] Minn/St.P - D.M.
[270] St.L - Chi
[270] Mad - Minn/St.P
[280] Chi - Det
[320] D.M. - St.L

✓ [80] Mad - Milw
✓ [95] Chi - Milw
✗ [150] Chi - Mad
✓ [235] Minn/St.P - D.M.
✓ [270] St.L - Chi
✓ [270] Mad - Minn/St.P
✓ [280] Chi - Det
✓ [320] D.M. - St.L

MST

	M/S.P.	DM	SL	C	Ma	Mi	D
Minn/St.Paul	0	1	0	0	1	0	0
Des Moines	1	0	0	0	0	0	0
St. Louis	0	0	0	1	0	0	0
Chicago	0	0	1	0	0	1	1
Madison	1	0	0	0	0	1	0
Milwaukee	0	0	0	1	1	0	0
Detroit	0	0	0	1	0	0	0

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



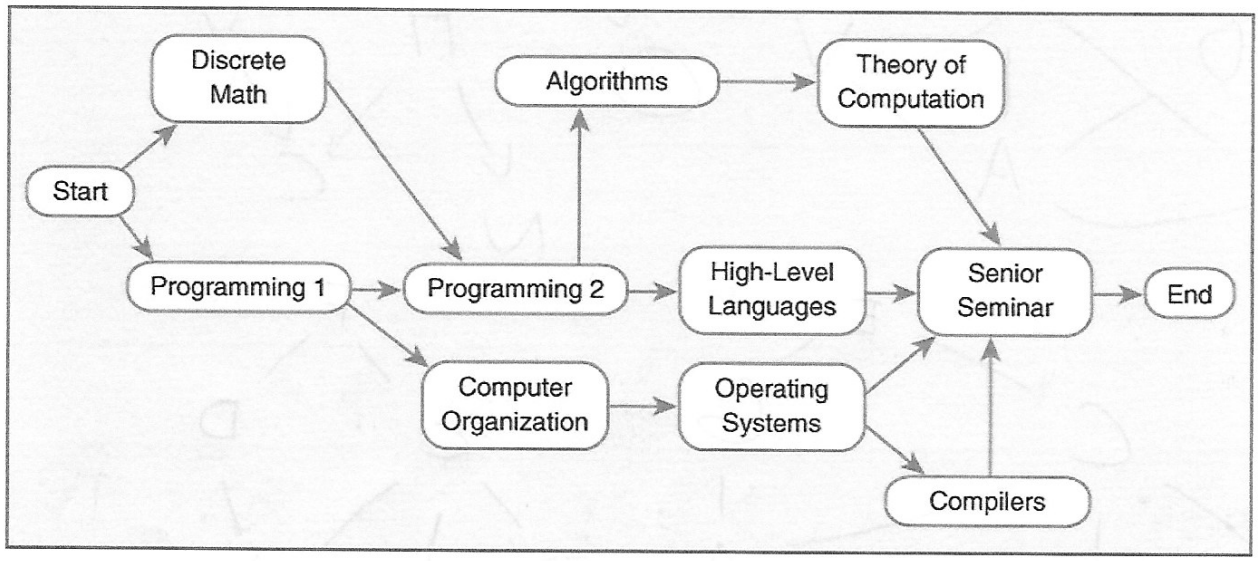
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
predCount	0	0	0	0	0	0	0	0	0	0

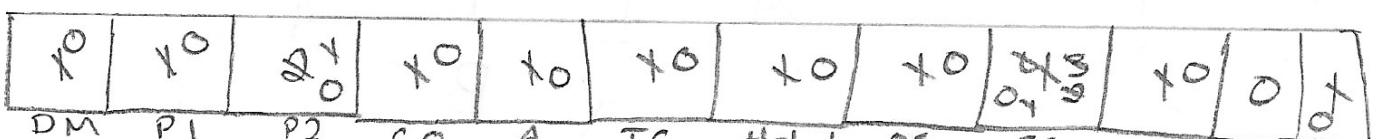
topOrder	0	7	1	2	5	6	4	8	3	9

queue | ~~0~~, ~~7~~, ~~1~~, ~~2~~, ~~5~~, ~~6~~, ~~4~~, ~~8~~, ~~3~~, ~~9~~

1 - Add 0 & 7 to queue
2 - Remove 0 from queue, add to topOrder, decrement its successors
Add 1 to queue
3 - Remove 7 from queue, add to topOrder, decrement successors
4 - Remove 1, add to topOrder, decrement successors
Add 2, 5, 6
5 - Remove 2, ...
Add 4
6 - Remove 5 ...
7 - Remove 6 ...
Add 8
8 - Remove 4
Add 3
9 - Remove 8
Add 9
10 - Remove 3
11 - Remove 9

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





Start, Discrete Math, P1, P2, CO, A, H-L L, OS, TC, C, SS, E

S, DM, P1, P2, CO, A, H-L L, OS, TC, C, SS, E

Start -> Discrete Math -> Programming 1 -> Programming 2 -> Computer Organization -> Algorithms -> High-Level Languages -> Operating Systems -> Theory of Computation -> Compilers -> Senior Seminar -> End