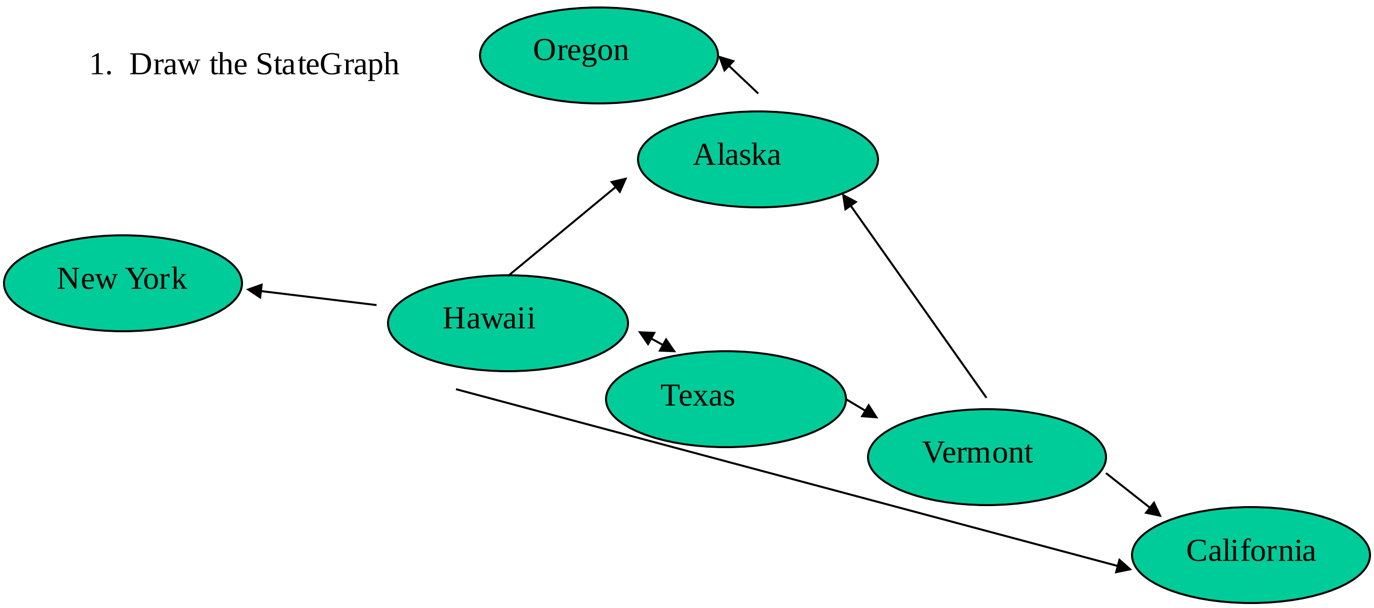
CMSC204
Kartchner

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)}

2. a. Is there a path from Oregon to any other state in the graph?
There is no path from Oregon to any other state.
b. Is there a path from Hawaii to every other state in the graph?
Yes, Hawaii can reach all other states.
c. From which state(s) in the graph is there a path to Hawaii?
Texas.

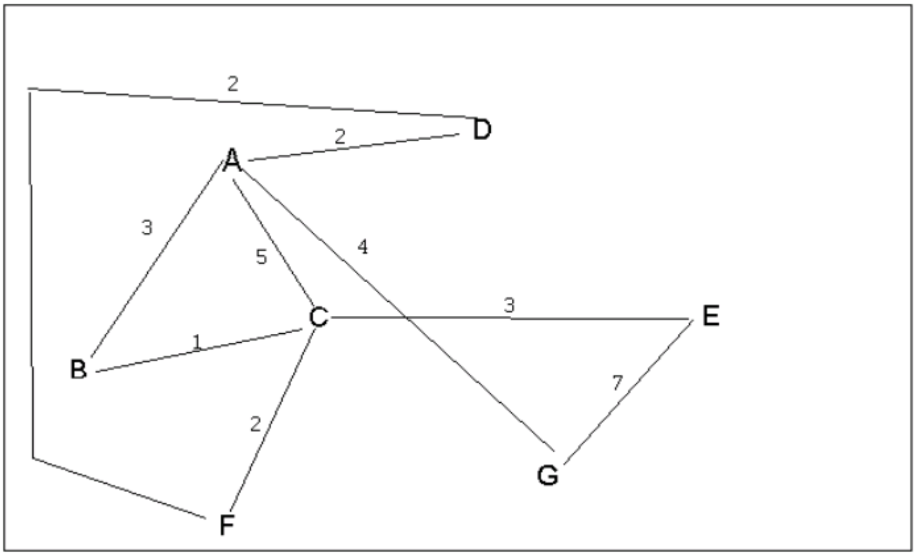
3. a. Show the adjacency matrix that would describe the edges in the graph.
Store the vertices in alphabetical order

States

States	Alaska	California	Hawaii	New York	Oregon	Texas	Vermont
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

Alaska: Oregon
California:
Hawaii: Alaska, California, New York, Texas
New York:
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

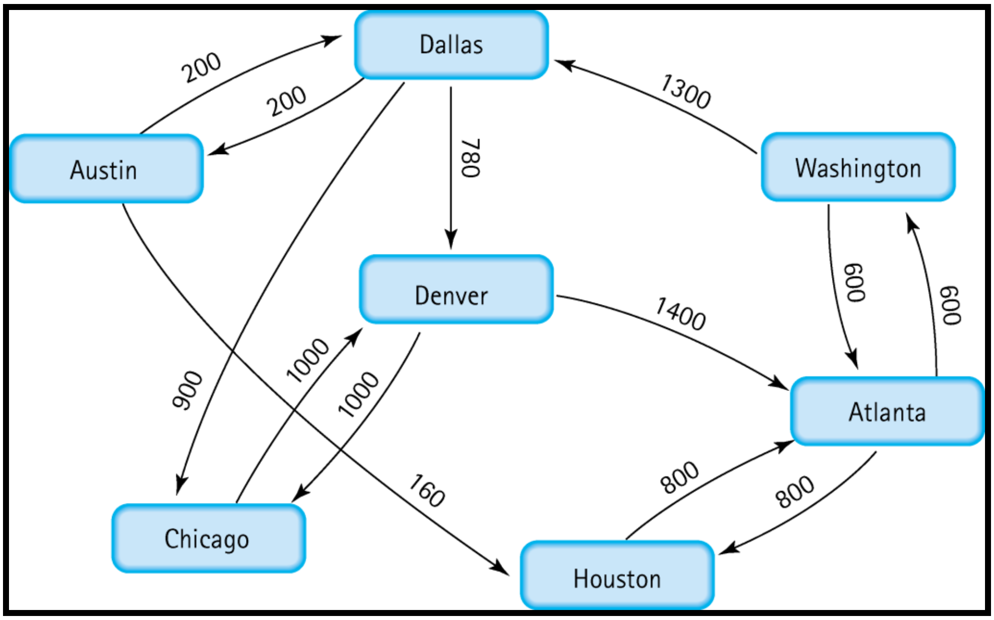
4 b. Which of the following lists the graph nodes in breadth first order beginning at F?

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

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

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

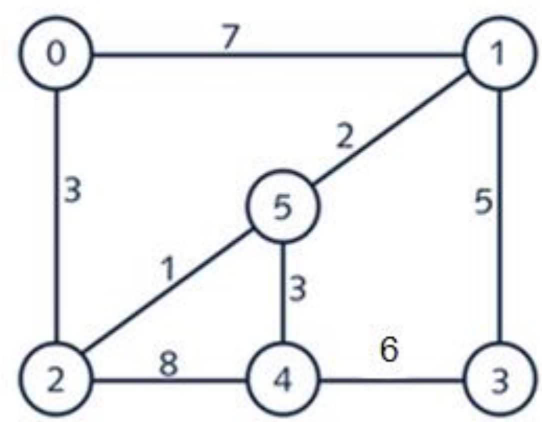
D) a, b, and c are all breadth first traversals



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

Atlanta → Atlanta: 0
Atlanta → Washington: 600
Atlanta → Houston: 800
Atlanta → Austin: 960
Atlanta → Dallas: 1160
Atlanta → Denver: 1940
Atlanta → Chicago: 2060

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



Prim's Algorithm Steps (Starting from 0):

1.Start: MST = {0}. Edges out: (0–1)=7, (0–2)=3. Choose (0–2)=3. MST: {0,2}.

2.Edges out: (0–1)=7, (2–4)=8, (2–5)=1. Choose (2–5)=1. MST: {0,2,5}.

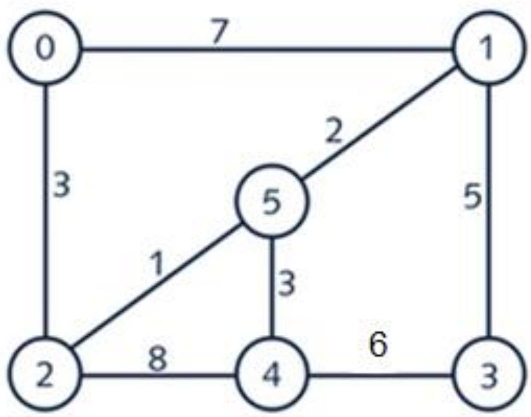
3.Edges out: (0–1)=7, (5–1)=2, (5–4)=3, (2–4)=8. Choose (5–1)=2. MST: {0,1,2,5}.

4.Edges out: (1–3)=5, (5–4)=3, (2–4)=8. Choose (5–4)=3. MST: {0,1,2,4,5}.

5.Edges out: (1–3)=5. Choose (1–3)=5. MST: {0,1,2,3,4,5}.

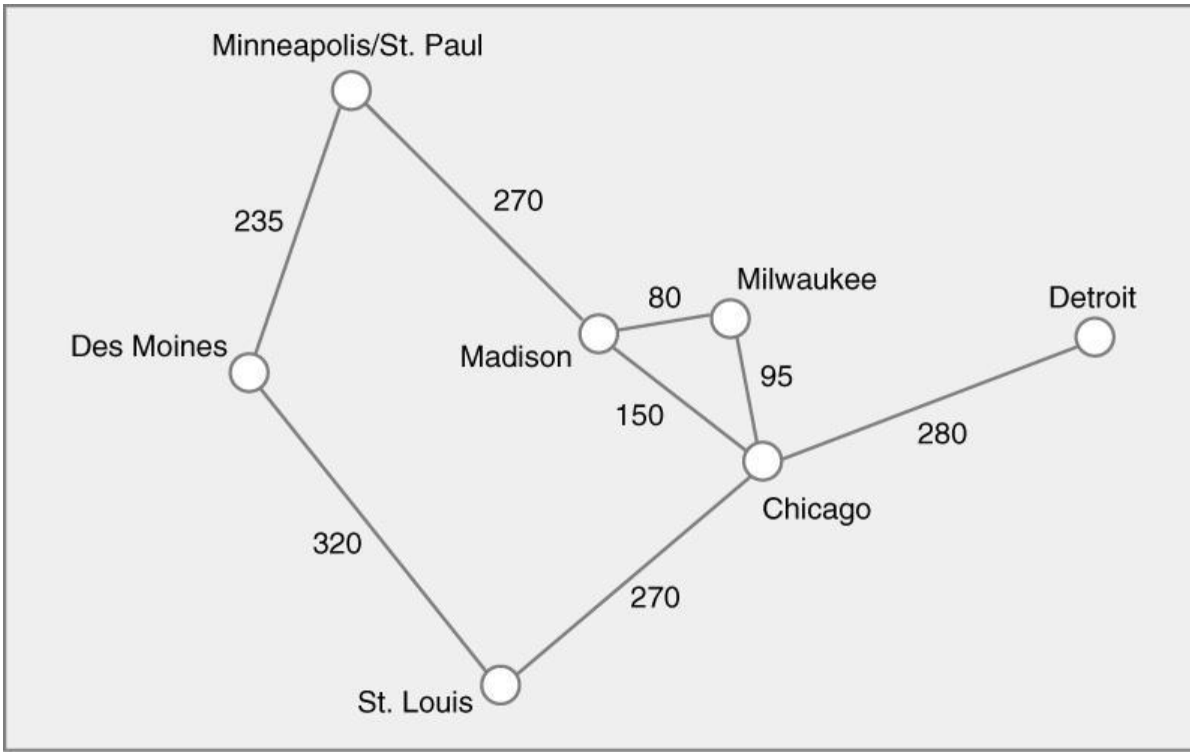
MST Edges: (0–2)=3, (2–5)=1, (5–1)=2, (5–4)=3, (1–3)=5. Total = 14.

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



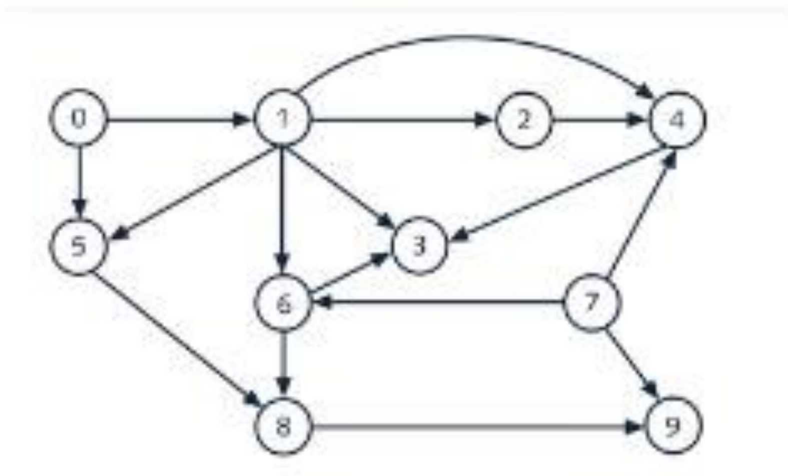
Edges sorted by weight:
(2–5)=1, (5–1)=2, (5–4)=3, (0–2)=3, (1–3)=5, (3–4)=6, (0–1)=7, (2–4)=8
Kruskal's Steps:
1.Pick (2–5)=1. MST: {(2–5)=1}
2.Pick (5–1)=2. MST: {(2–5)=1, (5–1)=2}
3.Pick (5–4)=3. MST: {(2–5)=1, (5–1)=2, (5–4)=3}
4.Pick (0–2)=3. MST: {(2–5)=1, (5–1)=2, (5–4)=3, (0–2)=3}
5.Pick (1–3)=5. MST: {(2–5)=1, (5–1)=2, (5–4)=3, (0–2)=3, (1–3)=5}
MST edges: (2–5), (5–1), (5–4), (0–2), (1–3)
Total weight: 1+2+3+3+5 = 14

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



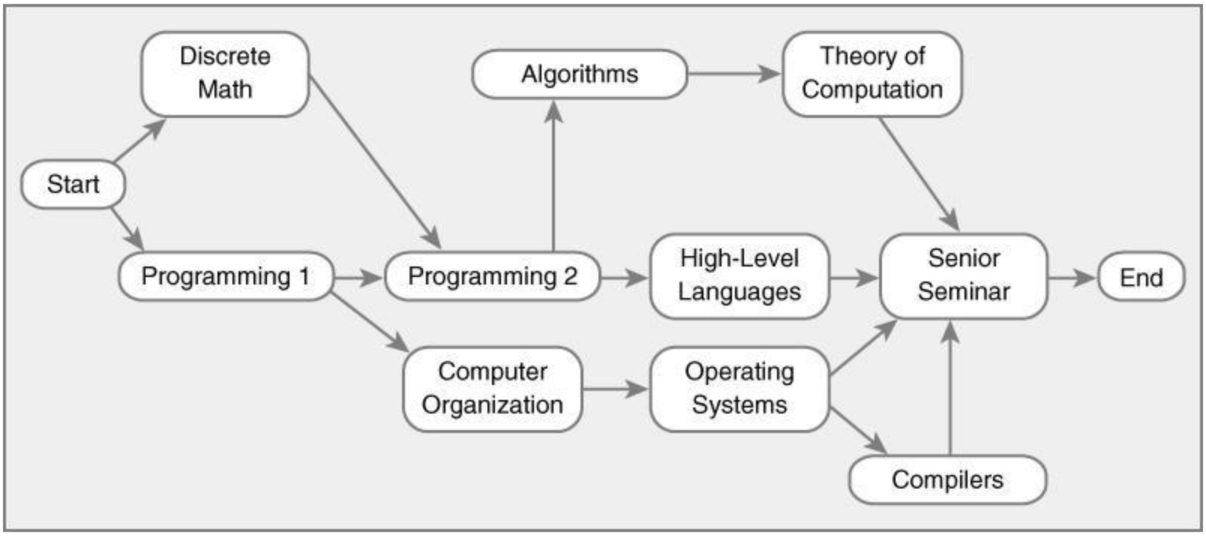
Prim’s algorithim;
1.From M/SP: pick (M/SP–Des Moines)=235
2.From M/SP, Des Moines: pick (M/SP–Madison)=270
3.From M/SP, Des Moines, Madison: pick (Madison–Milwaukee)=80
4.From M/SP, DM, Mad, Milw: pick (Milwaukee–Chicago)=95
5.From M/SP, DM, Mad, Milw, Chi: pick (Chicago–St. Louis)=270
6.From all above: pick (Chicago–Detroit)=280
MST Edges:
•M/SP–Des Moines (235)
•M/SP–Madison (270)
•Madison–Milwaukee (80)
•Milwaukee–Chicago (95)
•Chicago–St. Louis (270)
•Chicago–Detroit (280)
Total Weight: 235 + 270 + 80 + 95 + 270 + 280 = 1230

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



predCount:
•0:0
•1:1 (from 0)
•2:1 (from 1)
•3:3 (from 1,4,6)
•4:3 (from 1,2,7)
•5:2 (from 0,1)
•6:2 (from 1,7)
•7:0
•8:2 (from 5,6)
•9:2 (from 7,8)
Initialize Queue with in-degree 0: [0,7]
Process Queue:
•Dequeue 0: order=[0]. Decrease in-degree of 1(→0, enqueue), 5(→1). Queue=[7,1]
•Dequeue 7: order=[0,7]. Decrease in-degree of 6(→1), 9(→1), 4(→2). Queue=[1]
•Dequeue 1: order=[0,7,1]. Decrease in-degree of 2(→0, enqueue), 3(→2), 4(→1), 5(→0, enqueue), 6(→0, enqueue). Queue=[2,5,6]
•Dequeue 2: order=[0,7,1,2]. Decrease in-degree of 4(→0, enqueue). Queue=[5,6,4]
•Dequeue 5: order=[0,7,1,2,5]. Decrease in-degree of 8(→1). Queue=[6,4]
•Dequeue 6: order=[0,7,1,2,5,6]. Decrease in-degree of 3(→1), 8(→0, enqueue). Queue=[4,8]
•Dequeue 4: order=[0,7,1,2,5,6,4]. Decrease in-degree of 3(→0, enqueue). Queue=[8,3]
•Dequeue 8: order=[0,7,1,2,5,6,4,8]. Decrease in-degree of 9(→0, enqueue). Queue=[3,9]
•Dequeue 3: order=[0,7,1,2,5,6,4,8,3]. Queue=[9]
•Dequeue 9: order=[0,7,1,2,5,6,4,8,3,9]. Queue=[]
Topological Order:
[0, 7, 1, 2, 5, 6, 4, 8, 3, 9]

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



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