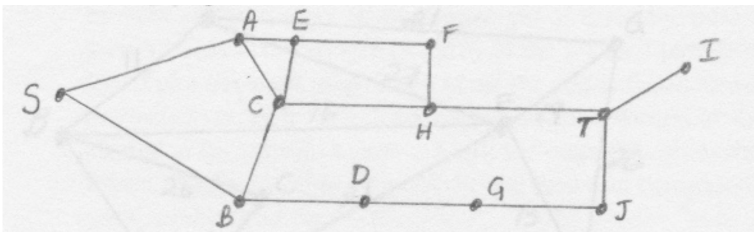
Math 272 Discrete Mathematics
Worksheet for Lesson 4: Shortest Paths in a Graph

(Some of the examples below are modified from *Contemporary Mathematics in Context*.)

1. The **length of a path in a graph** is the number of edges in the path. The distance between two vertices U and V in a graph is the length of the shortest path between U and V. Let G be the following graph:



(a) What is the length of the path SACBDGJT?

(b) There are two shortest paths from S to T. What are they? What is the length of each?

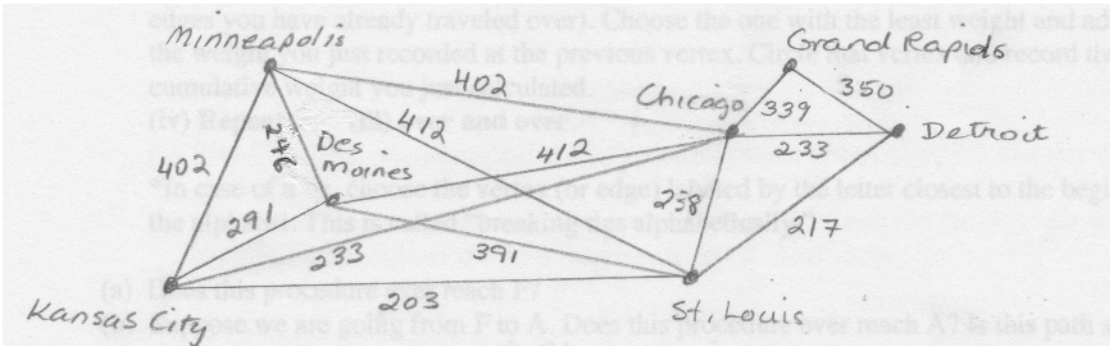
(c) Describe in words the procedure you used to find the shortest paths in (b).

(d) Finding all possible paths between S and T and checking their lengths could be a huge undertaking for a large graph. Invent a step-by-step procedure starting from S that would avoid having to check the length of every path from S to T.

(e) If you keep your procedure in (d) going past T until all vertices are reached, will you have found the shortest path from S to all the vertices in G (without having to go back and start from S again for each vertex)? The Breadth First Search does this. See the solutions.

2. Weighted graphs. A weighted graph is a graph with numerical values assigned to each edge (such as a road map with mileage between cities). The **length of a path in a weighted graph** (also called the weight of the path) is the sum of the values assigned to the edges of the path.

The following weighted graph shows airfares charged by a major airline for round-trip tickets around the Midwest with a Saturday night stay-over.



(a) What does the shortest path between two vertices tell you about airfares?

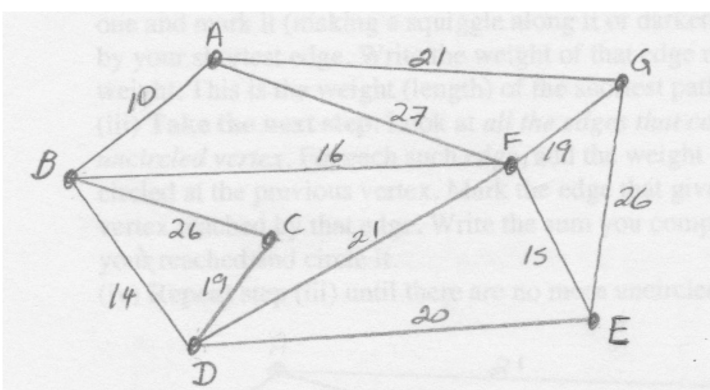
(b) What is the cheapest airfare between Minneapolis and Detroit? Between Kansas City and Grand Rapids?

(c) Why do you think Chicago and St. Louis are called "major hubs"?

(d) If your friend asks your advice about flying from Minneapolis or Kansas City to Detroit, what would you tell them?

3. Shortest paths in weighted graphs. Weighted graphs are good models for many different network problems.

Seven small towns in Jefferson County are connected by gravel roads. The distances are in miles. (Not drawn to scale and the roads are often curved.)



(a) What is the shortest path from town A to town E? What is its length?

(b) What is the shortest path from town F to town A? What is its length?

Once again, trying all possible paths from F to A and choosing the shortest could be very time consuming if the graph were bigger than this one. Let's try to come up with an algorithm.

4**. "Nearest Neighbor Algorithm."** Here is one possible algorithm, which we could call the "Nearest Neighbor Algorithm". Suppose we are trying to find the shortest path from vertex F to vertex A in the above graph.

> **Step 1. Start somewhere**. Find the starting vertex and circle it. Since it is the start, write the number zero next to it.
>
> **Step 2. Move to another vertex**. Look at all the edges incident to your starting vertex. Choose the edge with the least weight and mark it (making a squiggle along it or darkening it will do). (In case of a tie, choose the edge leading to the vertex labeled by the letter closest to the beginning of the alphabet. This is called "breaking ties alphabetically.") Circle the vertex reached by your shortest edge. Write the weight of that edge next to the vertex you reached.
>
> **Step 3. Move to another vertex.** Look at all the edges incident to the vertex you just reached (except any edges you have already traveled over). Choose the one with the least weight (breaking ties alphabetically) and add that to the weight you just recorded at the previous vertex. Circle that vertex and record next to it the cumulative weight you just calculated.
>
> **Step 4. Repeat Step 3 over and over**.

(a) Does this procedure ever reach A? What is the total distance of this path? Does this path give the shortest distance from F to A?

(b) Suppose we are going from A to F instead. Does this procedure ever reach F? Is this path the reverse of the path you found in (a)? Does this procedure give the shortest path from A to F? Why or why not? Give clear reasons.

This is why the "Nearest Neighbor" algorithm is no use at all for finding shortest paths.

5**. Dijkstra's Algorithm**. How can we fix what went wrong? E.W. Dijkstra, a Dutch mathematician, discovered an efficient algorithm in 1959. It finds the shortest distance from one vertex to all other vertices in the graph. The idea is simple.

Again, break ties alphabetically.

Step 1. Start somewhere. Find your starting vertex V and circle it. Since it is the start, write the number zero next to it. Circle the zero.

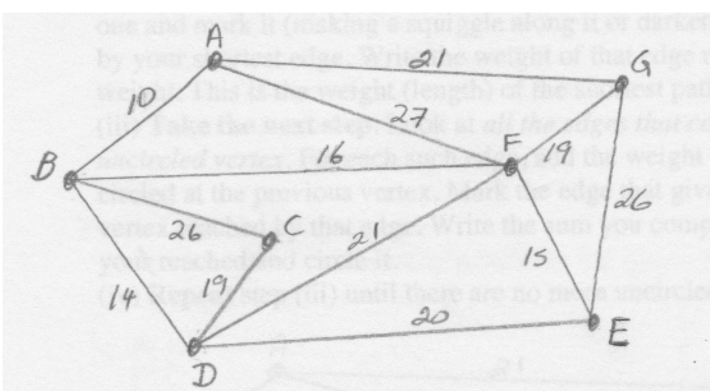
Step 2. Choose the next vertex to move to. Look at all the edges incident to your starting vertex. Choose the shortest one and mark it (making a squiggle along it or darkening it will do). Circle the vertex reached by your shortest edge. Write the weight of that edge, together with its predecessor, next to that new vertex and circle them. This is the weight (length) of the shortest path from the start to that vertex.

Step 3. Choose the next vertex to move to. Look at ***all the edges that connect any circled vertex with any un-circled vertex***. For each such edge, add the weight of this edge to the weight circled at the previous vertex and record it next to the un-circled vertex you have just reached, together with its predecessor. This is the total distance from the start to that un-circled vertex via that predecessor vertex. Do NOT circle it. NOW. . . examine all these un-circled vertices you have just reached and choose the one with the smallest total distance. Circle that vertex and mark the corresponding edge. Also circle the sum you computed for that edge together with its predecessor.

Erase any un-circled numbers and predecessor notes you made at this step in preparation for the next step.

Step 4. Repeat Step 3 over and over until there are no more un-circled vertices.

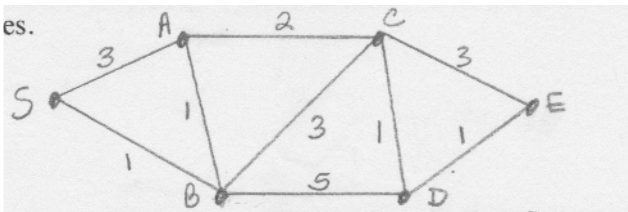
Step 5. Work backwards through the circled predecessor vertices to get the shortest path from the starting vertex to any vertex in the graph.



(a) Apply Dijkstra's Algorithm to the road map above starting at A. Convince yourself that it gives the shortest route from A to each of the other towns and that the circled weight next to each vertex gives the shortest distance from A to that town. Some copies of this graph are supplied with this worksheet for you to mark your steps on.

(b) What is the difference between Dijkstra's Algorithm and the Nearest Neighbor algorithm? Why does Dijkstra's work and the Nearest Neighbor not work?

6. **Another example**. Imagine the vertices in the following graph are art galleries and the edges are roads with distances in miles.



(a) Use Dijkstra's Algorithm to find the shortest distance from gallery S to every other art gallery. Copies of this graph can be found at the end of this worksheet to allow you to do the steps.

(b) Construct a matrix of shortest distances as follows. The entry in row A and column D is the shortest distance between A and D. And so on.

	S	A	B	C	D	E
S	0	2	1	4	5	6
A	2	0	1	2	3	4
B	1	1	0	3	4	5
C	4	2	3	0	1	2
D	5	3	4	1	0	1
E	6	4	5	2	1	0

The top row of the matrix consists of the answers you got in (a) above. Explain how you can use these values in the top row to find all the rest of the values in the matrix.

(c) What is the farthest you have to go to get from one gallery to any other gallery?

(d) For each gallery, compare its row with its column. What pattern do you see? Explain why this pattern should be expected.

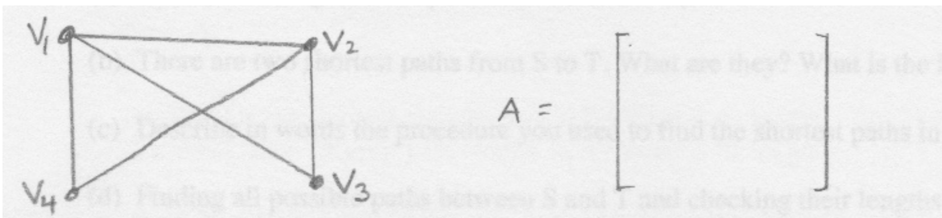
(e) The row sum for gallery A is less than the row sum for gallery B. what does this mean in terms of galleries and distances?

(f) Which galleries are the most isolated?

(g) Suppose a store selling art supplies is to be added on to one of the galleries. At which gallery should it be built? Why?

7. Counting the Number of Paths between Vertices: **Powers of adjacency matrices**.

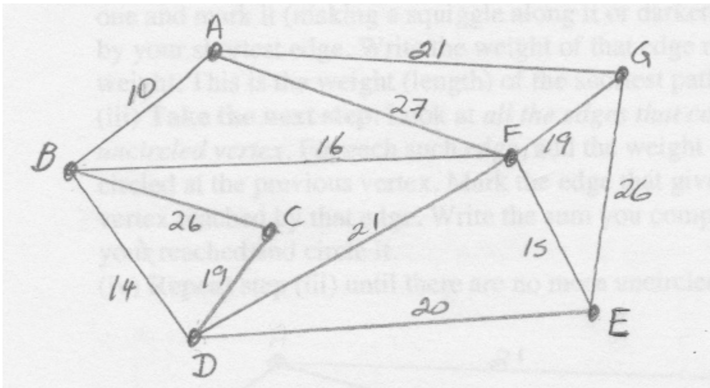
(a) Write down the adjacency matrix for the graph. Note that the ij-entry gives the number of edges joining V_i and V_j.

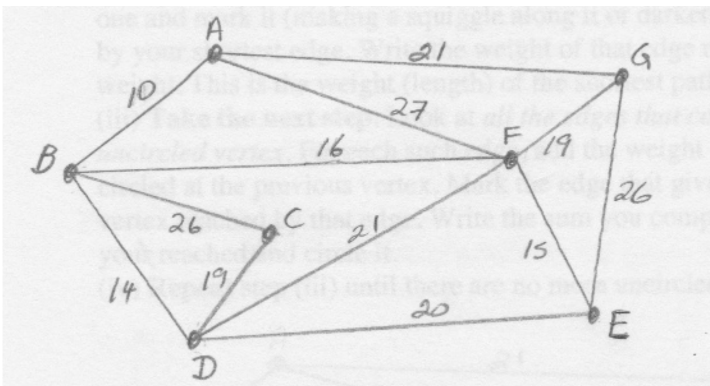


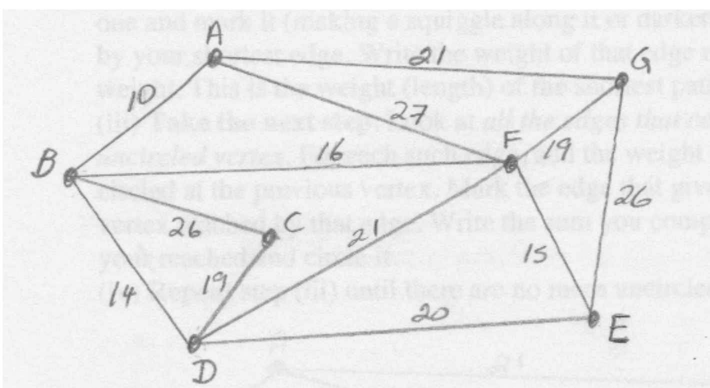
(b) Look carefully at the graph and determine how many paths of length 2 there are joining V_1 and V_4. Remember that you are allowed to repeat edges. Do the same for V_3 and V_4. Now find the number of paths of length 2 beginning and ending at V_1.

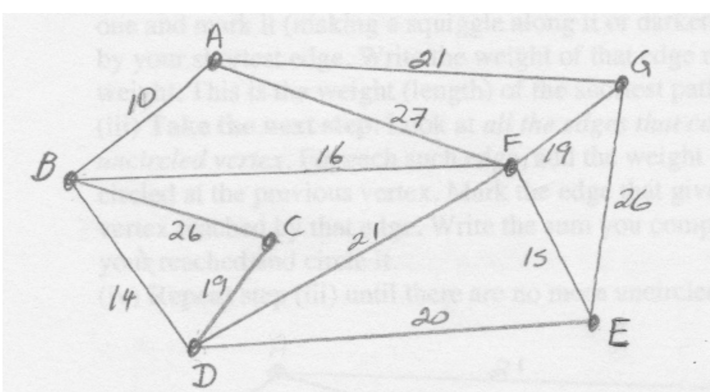
(c) Calculate A^2 (that is, A times A using matrix multiplication). What do you notice about the 1–4 entry, the 3–4 entry, and the 1–1 entry if you compare them to your answers to (b) above? Why should you expect this to be so? What can you say in general about A^2 in this regard?

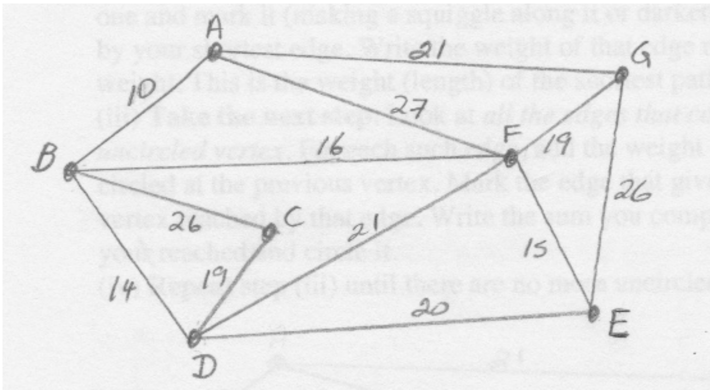
(d) What do you expect the entries in A^3 to tell you? (A^3 is A times A^2.) Why?

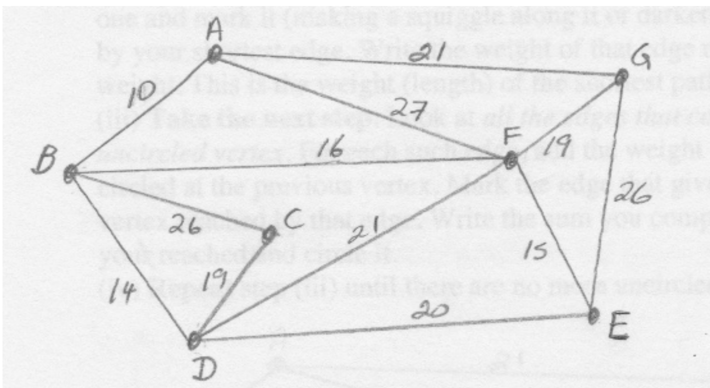

A
G
10
21
27
B
16
F
19
26
26
C
21
15
14
19
20
E
D

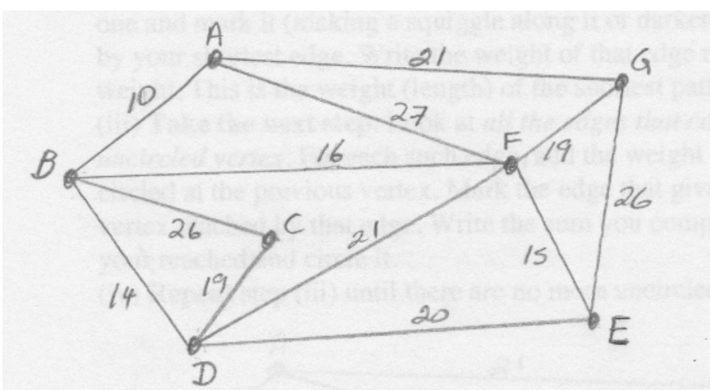

A
G
10
21
27
B
16
F
19
26
26
C
21
15
14
19
20
E
D

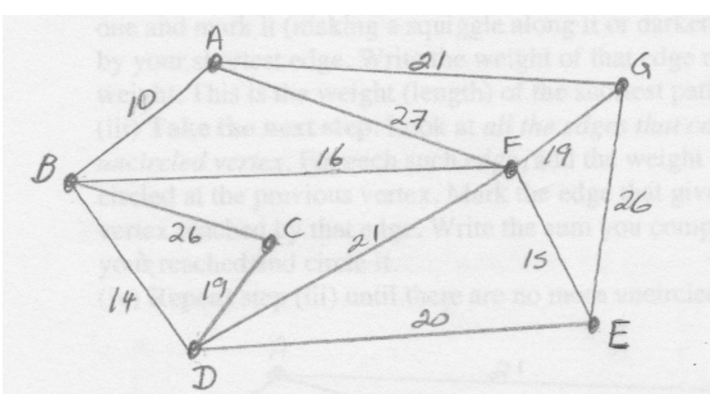

A
G
10
21
27
B
16
F
19
26
26
C
21
15
14
19
20
E
D

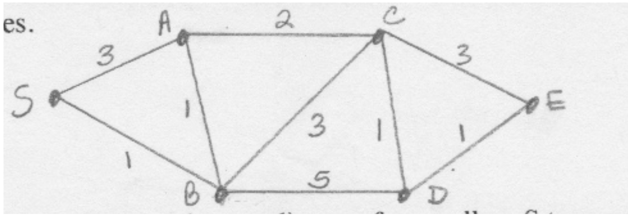

A
G
10
21
27
B
16
F
19
26
26
C
21
15
14
19
20
E
D

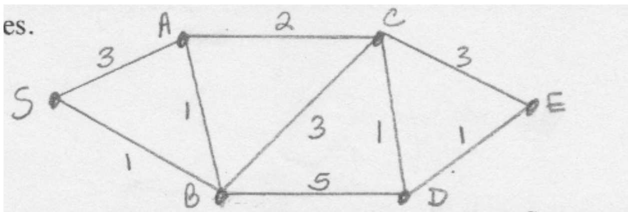

A
G
B
F
C
D
E
10
21
27
16
19
26
26
21
15
14
19
20

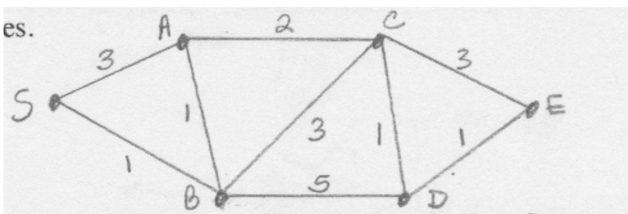

A
G
B
F
C
D
E
10
21
27
16
19
26
26
21
15
14
19
20

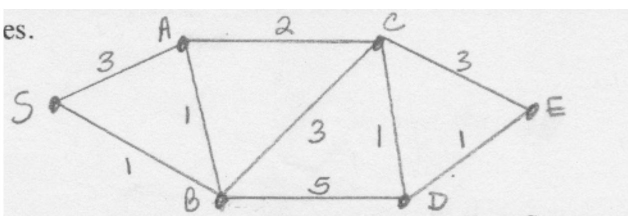

A
G
B
F
C
D
E
10
21
27
16
19
26
26
21
15
14
19
20

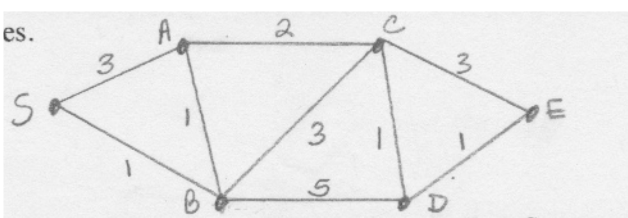

A
G
B
F
C
D
E
10
21
27
16
19
26
26
21
15
14
19
20

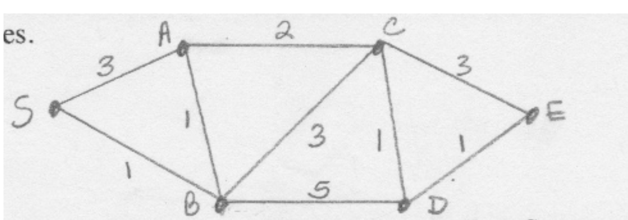

es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D

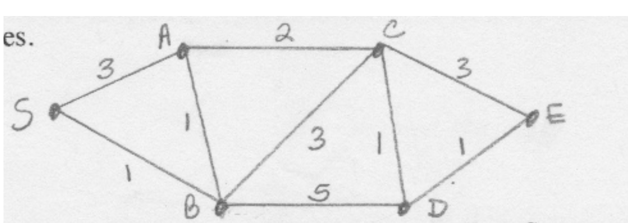

es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D


es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D


es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D


es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D


es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D


es.
A
2
C
3
3
S
E
1
3
1
1
1
B
5
D