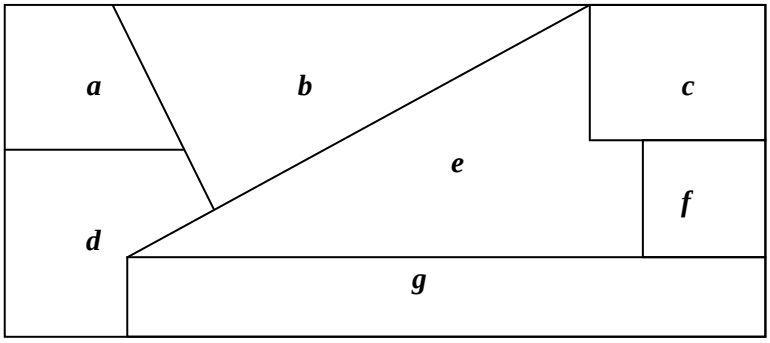
ECS 20 **Homework #6, Graph Theory (50 points)** **Winter 2012**

Due: Tuesday, February 28th by 4:00pm

1. (4 points) Provide the result of the final operation after each series of operations is executed.
   a) On a stack: push(7), push(12), pop(), push(15), top(), push(9), pop(), push(5), top(), pop(), top().
   b) On a queue: enqueue(7), enqueue(12), dequeue(), enqueue(15), front(), enqueue(9), dequeue(), enqueue(5), back(), dequeue(), front().
   c) On a minimum priority queue: insert(7), insert(12), deleteMin(), insert(15), findMin(), insert(9), deleteMin(), insert(5), findMin(), deleteMin(), findMin().
   d) On a maxmum priority queue: insert(7), insert(12), deleteMax(), insert(15), insert(9), deleteMax(), insert(5), findMax(), deleteMax(), findMax().

2. (5 points) Is it possible in a group of nine people for each to be friends with exactly five others? Justify your answer.

3. (5 points) Imagine the diagram shown below is a map with countries labeled *a-g*. Is it possible to color the map with only three colors so that no two adjacent countries have the same color? To answer this question, draw and analyze a graph in which each country is represented by a vertex and two vertices are connected by an edge i, and only if, the countries share a common border.

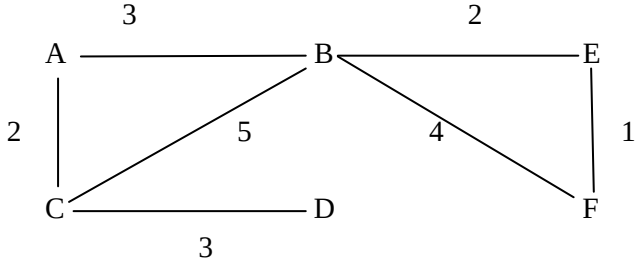


4. (5 points) Given the following table of airfares between cities, what is the route that will permit the traveling saleswoman to visit all of the cities, and return to her starting city, Seattle, spending the least amount of money? What route would be found using the Nearest Neighbor Algorithm starting at Seattle?

City	Seattle	Boston	New York	New Orleans	Phoenix
Seattle		409	389	429	119
Boston	409		109	239	379
New York	389	109		229	319
New Orleans	429	239	229		309
Phoenix	119	379	319	309	

5. (5points) Prove that if G is a simple graph with at least two vertices, then G has at least two vertices that have the same degree. Hint: In any simple graph with *n* vertices, there cannot be both a vertex of degree 0 and a vertex of degree *n* – 1.

6. (26 points) Answer the questions based on the following weighted graph, G.



a) What is the set V?
b) What is |V|?
c) What is the set E?
d) What is |E|?
e) Is this a sparse graph or dense graph? Justify your answer.
f) Is this a simple graph or a multigraph? Justify your answer.
g) Are B and F adjacent?
h) What are the degrees of all the vertices?
i) What is the sum of the degrees of the whole graph?
j) List all the simple paths from D to F.
k) List all cycles in G.
l) List the cutpoint(s) of G.
m) Is G traversable? Justify your answer.
n) Is G a Eulerian graph? Justify your answer.
o) Which edge(s) would you need to add to make a Hamiltonian circuit possible? List a path of your proposed Hamiltonian circuit.
p) With the addition of the edge(s) in part o), would this graph be suitable for the Traveling Salesman Problem? Justify your answer.
q) Assume that F is the distinguished vertex, use Dijkstra's algorithm to fill in a table like the one below to determine the shortest paths from F.

Vertex	Known	dv	pv
A			
B			
C			
D			
E			
F			

r) Based on your table in the previous question, what is the shortest path from F to D?
s) Is G a bipartite graph? If not, then justify your answer. If it is, then provide the two disjoint subsets of the vertices.
t) Assume that D is the vertex with which to start the tree, use Prim's algorithm to fill in a table like the one in part q) to determine the minimum spanning tree for G.
u) Based on your table in the previous question, what are the edges in the minimum spanning tree of G?
v) What is/are the region(s) of G? What is the degree of each region?
w) Use Welch and Powell's Coloring Algorithm to color G. List the colors of each vertex, with red being the first color used, white the second, blue the third, and green the fourth. You need only provide one coloring solution.
x) Starting at D list the vertices as they are encountered in a depth first search. When confronted with multiple vertices, process them in alphabetical order.
y) Starting at F list the vertices as they are encountered in a breadth first search. When confronted with multiple vertices, process them in alphabetical order.
z) Create an adjacency matrix for G.