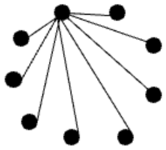
Homework #11

Problem 1 [20 pts (10, 10)]**: Graph Theory**

1. Nine people are at a party. Explain why it is impossible for each of the nine party goers to have shaken hands with exactly five other party goers.
2. If a graph with 1 million vertices consists of 3 connected components, what is the minimum number of edges it might have? Hint: How do we ”minimally connect” a graph?

Solution:

1. Consider 9 party goers to stand in a circle and construct a k9 graph.
   To disconnect one of the vertices, at least 8 edges need to be removed to make it disconnected from the k9 graph.
   So each of the nine party goers must have shaken hands with exactly 8 other party goers rather that 5.



2. If we are looking for the minimum number of edges a group of vertices that are connected. These vertices must line up and each vertex must have degree of 2 except for vertices on both ends which only have a degree of 1.

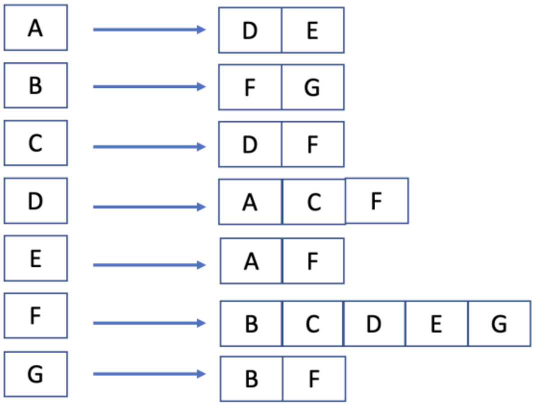
   To disconnect the graph and split it into three connected components, only two edges need to be removed to make it.

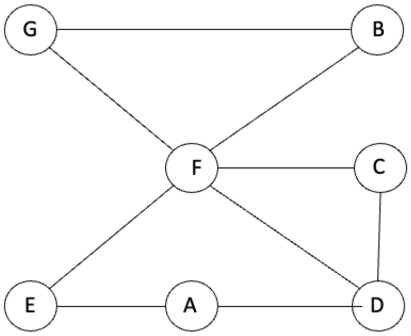
   In this sense, the minimum number of edges must be 1,000,000 – 2= 999,998 edges to form 3 connected components.

Problem 2 [20 pts (10,10): **Graph Representations**

$$\begin{array}{cccccccc} & A & B & C & D & E & F & G \\ A & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ D & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ E & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ F & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ G & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

For the above adjacency matrix representation of an unweighted undirected graph, create an equivalent adjacency list representation and draw the resulting graph.





Deg(A) = 2

Deg(B) = 2

Deg(C) = 2

Deg(D) = 3

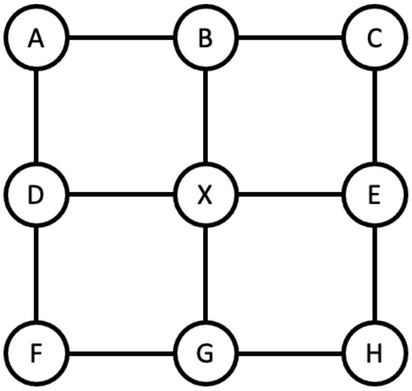
Deg(E) = 2

Deg(F) = 5

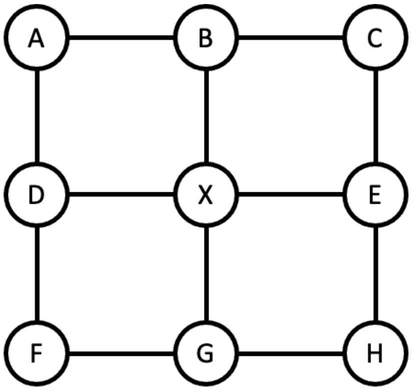
Deg(G) = 2

Problem 3 [20 pts (10,10)**: Graph Traversal**

For the 9-vertex graph below, highlight the edges that would be traversed using Depth-First Search (left) and Breadth-First Search (right) starting at the middle vertex labeled *X*. Add a number, 1 to 9 next to each vertex to show the order in which each vertex is visited, starting with a 1 next to *X*. In cases of a tie, the traversal should proceed alphabetically.



Depth-First Search



Breadth-First Search

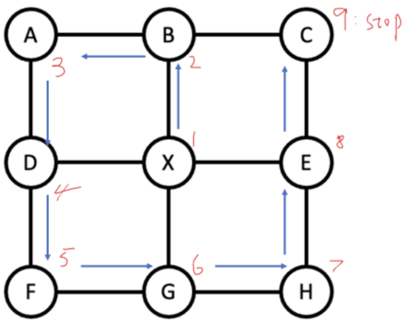
Solution:

DFS:

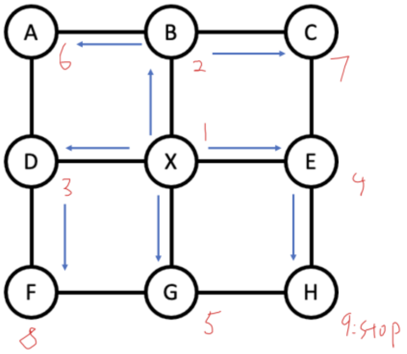
X -> B -> A -> D -> F ->G ->H ->E -> C

BFS:

X -> B -> D -> G -> E -> A -> C -> F -> H



Depth-First Search



Breadth-First Search