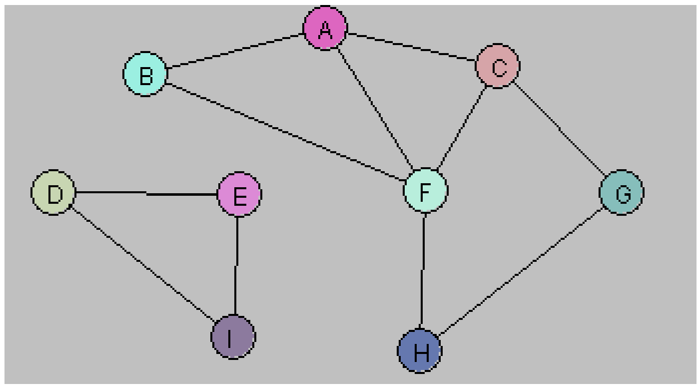
Lab 10
Due Friday 2 PM

1. *Induced Graphs.* Answer questions about the graph G = (V, E) displayed below.

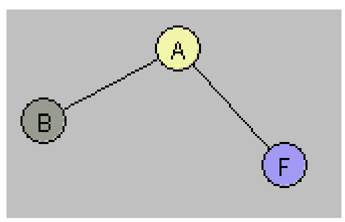


   A. Let U = {A, B}. Draw G[U].

   B. Let W = {A, C, G, F}. Draw G[W].

   C. Let Y = {A, B, D, E}. Draw G[Y].

   D. Consider the following subgraph H of G:



   Is there a subset X of the vertex set V so that H = G[X]? Explain.

   E. Find a way to partition the vertex set V into two subsets V_1, V_2 so that each of the induced graphs $G[V_1]$ and $G[V_2]$ is connected and $G = G[V_1] \cup G[V_2]$.

2. *Graph Implementation.* Use the BFS class to solve the following problems. Implement by implementing the unimplemented methods in the Graph class.

- Given two vertices, is there a path that joins them?
- Is the graph connected? If not, how many connected components does it have?
- Does the graph contain a cycle?

Hint: For the third problem, you are allowed to use the following Fact (which we will prove in tomorrow's class):

> **Fact**: Suppose G is a graph with *n* vertices and *m* edges. Let F be a spanning forest in G (that is, F is the subgraph of G that is obtained by running the Find Spanning Tree algorithm on G; if G is connected, F will be a spanning tree; otherwise it will be a union of spanning trees, one inside each connected component of G). Let m_F be the number of edges in F. Then G contains a cycle if and only if $m > m_F$.

3. *Graph Exercises.*

A. Suppose G = (V, E) is a connected simple graph. Suppose $V_1, V_2, \ldots, V_k$ are disjoint subsets of V and that $V_1 \cup V_2 \cup \ldots \cup V_k = V$. Show that there is an edge (x,y) in E such that for some $i \neq j$, x is in V_i and y is in V_j.

B. In class it was shown that a graph G = (V, E) is connected whenever the following is true,

$$(*) \qquad \epsilon > \binom{\nu - 1}{2}$$

where ν is the number of vertices and ϵ is the number of edges. Is the following true or false?

Every connected graph satisfies the inequality (*).

Prove your answer.

C. Suppose G is a graph with two vertices. What is the minimum number of edges it must have in order to be a connected graph? Suppose instead G has three vertices; what is the minimum number of edges it must have in order to be connected? Fill in the blank with a reasonable conjecture:

If G has n vertices, G must have at least ____ edges in order to be connected.

4. Show that for any simple graph G (not necessarily connected) having n vertices and m edges, if $m \geq n$, then G contains a cycle.

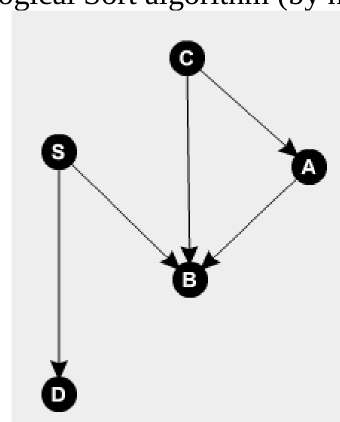
5. Suppose G = (V, E) is a connected simple graph. Suppose $S = (V_S, E_S)$ and $T = (V_T, E_T)$ are subtrees of G with no vertices in common (in other words, V_S and V_T are disjoint). Show that for any edge (x,y) in E for which x is in V_S and y is in V_T, the subgraph obtained by forming the union of S, T and the edge (x,y) (namely, $U = (V_S \cup V_T, E_S \cup E_T \cup \{(x,y)\})$) is also a tree.

6. Implement a subclass ShortestPathLength of BreadthFirstSearch that will provide, for any two vertices x, y in a graph G, the length of the shortest path from x to y in G, or -1 if there is no path from x to y. Use the ideas mentioned in the slides for your implementation. Be sure to add a method of the Graph class having the following signature:

```
int shortestPathLength(Vertex u, Vertex v)
```

which will make use of your new subclass

7. Perform the General Topological Sort algorithm (by hand) on the following graph.



In the slides, the starting vertex was S, and then to complete the algorithm, C was the second starting point. Show what happens if now, the first vertex is C, and then the second one is S.