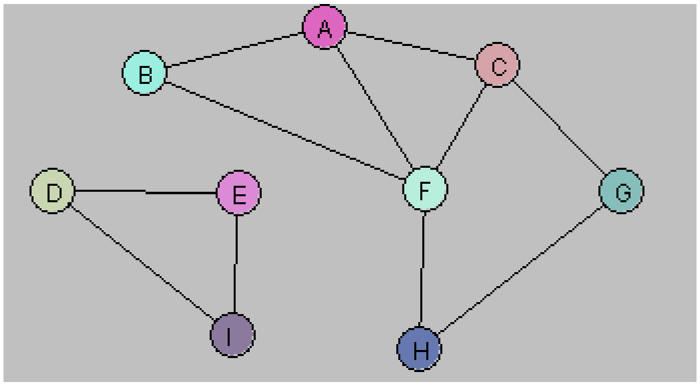
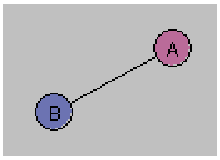
Lab 10
Due Thursday 2 PM

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



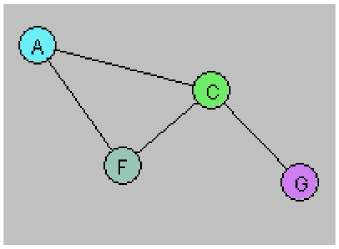
   A. Let U = {A, B}. Draw G[U].

   Solution:



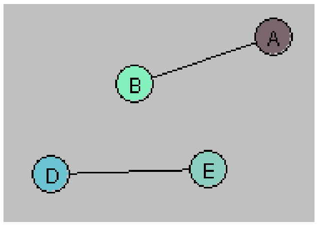
   B. Let W = {A, C, G, F}. Draw G[W].

   Solution:

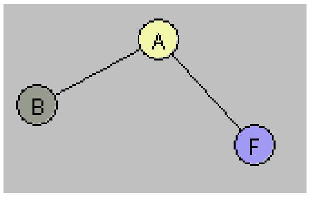


   C. Let Y = {A, B, D, E}. Draw G[Y].

   Solution:

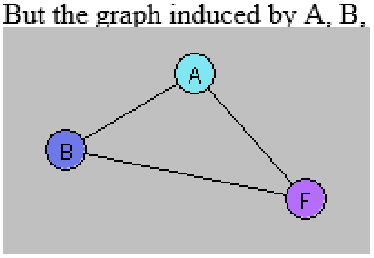


D. Consider the following subgraph H of G:



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

Solution. Any such X would have to contain the vertices A, B, F, and no others. But the graph induced by A, B, F is the following, which is not the same as H.



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 $G = G[V_1] \cup G[V_2]$.

Solution: $V_1 = \{D, E, I\}$ and $V_2 = \{A, B, C, F, G, H\}$.

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?

Solution: See the zipped graph package in this directory: `labsolns10`

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.

Solution: Let u be in V_1 and let v be in V_2. Let p be a path in G from u to v. Let y be the first vertex in p that is not in V_1; y must belong to V_j for some $j \neq 1$. Let x be the immediate predecessor of y in p; note x belongs to V_1. The edge in p that joins x and y is the desired edge.

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

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

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

Every connected graph satisfies the inequality (*).

Prove your answer.

Solution. A graph of the form O – O – O – O with 4 vertices provides a counterexample.

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 <u>n - 1</u> 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.

Solution. It is enough to show that if G contains no cycle, then $m < n$. We have already proven this result when G is connected. Suppose G is not connected with components $G_1, G_2, \ldots, G_k$, with $k > 1$. Since G contains no cycle, none of the components contains a cycle either; therefore, for each i, we have that G_i is a tree. Let m_i denote the number of edges in G_i and n_i the number of vertices in G_i. Then for each i, $m_i = n_{i\text{-}1}$. Therefore

$$m = m_1 + m_2 + \ldots + m_k = (n_1-1) + (n_2-1) + \ldots + (n_k-1) = (n_1+n_2+\ldots+n_k) - k = n-k < n-1 < n.$$

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.

   Solution. U is connected: If u, v are vertices in U, the only difficult case is the one in which u is in V_S and v is in V_T (or conversely). To find a path in that case, get a path p from u to x, another path q from y to v, and the combined path p U {(x,y)} U q is the needed path from u to v in U.

   U contains no cycle: It suffices to show that U has the right number of edges. Let $n_S = |V_S|$, $m_S = |E_S|$, and $n_T = |V_T|$, $m_T = |E_T|$. Then U has $n_S + n_T$ vertices. We show U has $n_S + n_T - 1$ edges. Let m be the number of edges in U. Then

   $$m = m_S + m_T + 1 = (n_S - 1) + (n_T - 1) + 1 = n_S + n_T - 1 = n - 1,$$

   as required.

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

   Solution: See the zipped graph package in this directory: `labsolns10`

7. True or False: Every disconnected graph is sparse. Prove your answer.

   Solution: False. Let n be any natural number (intuitively, n is big). Let

   $$G = K_{n-1} \cup K_1$$

   Then G has n vertices and the number of edges m in G is precisely the number of edges in K_{n-1}, namely, $(n-1)(n-2)/2$, which belongs to $\Theta(n^2)$. Therefore, G is not sparse.