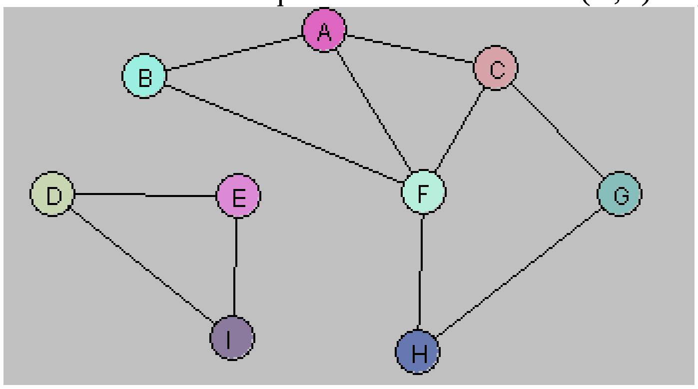
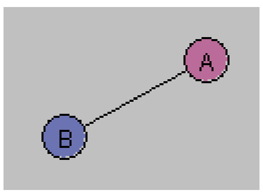
Lab 11 Solutions

1. **Problem 0.** Answer questions about the G = (V,E) displayed below.



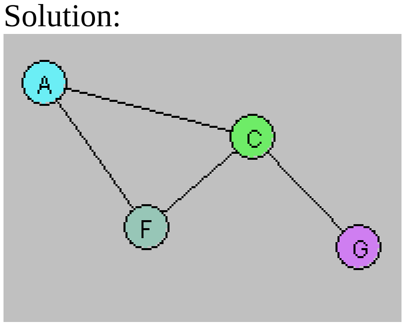
A. Let U = {A, B}. Draw G[U].

Solution:



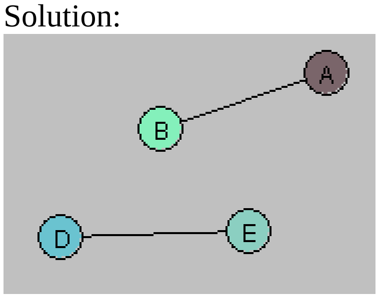
B. Let W = {A, C, G, F}. Draw G[W].

Solution:

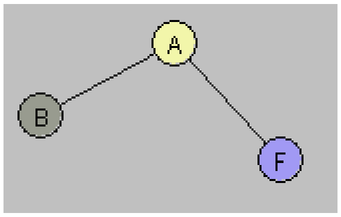


C. Let Y = {A, B, D, E}. Draw G[Y].

Solution:

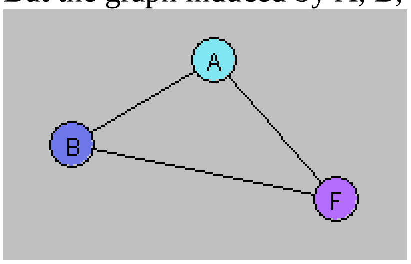


D. Consider the following subgraph H of G:



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

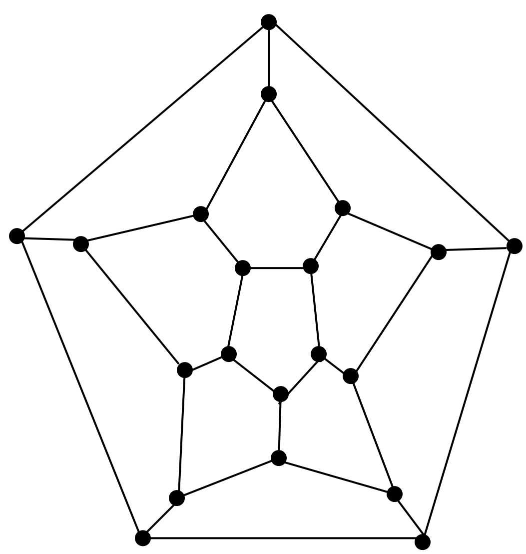
Solution. Any such X would have to contain the vertices A, B, F, and no others. But the graph induced by A, B, F is the following, which is not the same as H.



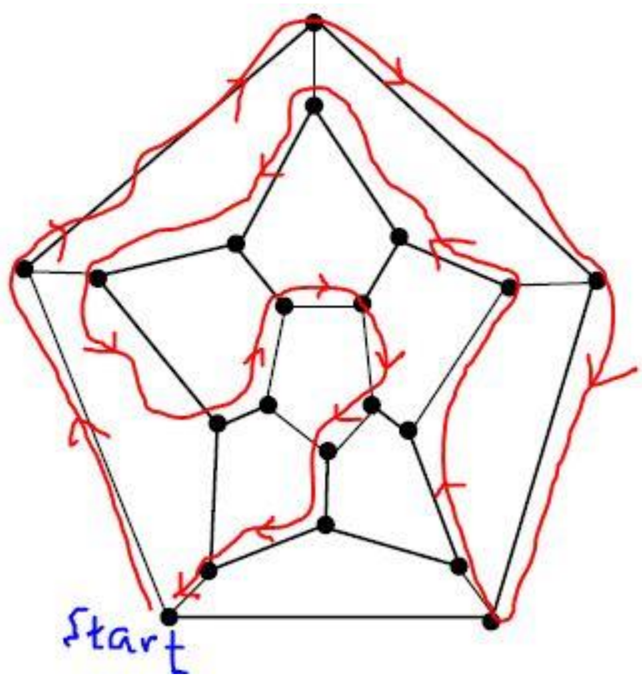
E. Find a way to partition the vertex set V into two subsets V_1, V_2 so that each of the induced graphs $G[V_1]$ and $G[V_2]$ is connected and $G = G[V_1] \cup G[V_2]$.

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

2. *Hamiltonian Graphs.* The following graph has a Hamiltonian cycle. Find it.



Solution:



3. Express in pseudo-code an algorithm which accepts as input a graph G and which outputs a vertex cover for G of smallest possible size. You may make use of the PowerSet algorithm without showing any pseudo-code details indicating how it works. Also, you may assume that your algorithm can make use of these operations freely:

    `computeEndpoints(e)` //returns the two endpoints of the edge e

    `belongsTo(x, U)` // returns true if vertex x belongs to set U; false otherwise

    Follow the rules for the pseudo-code language as completely as possible.

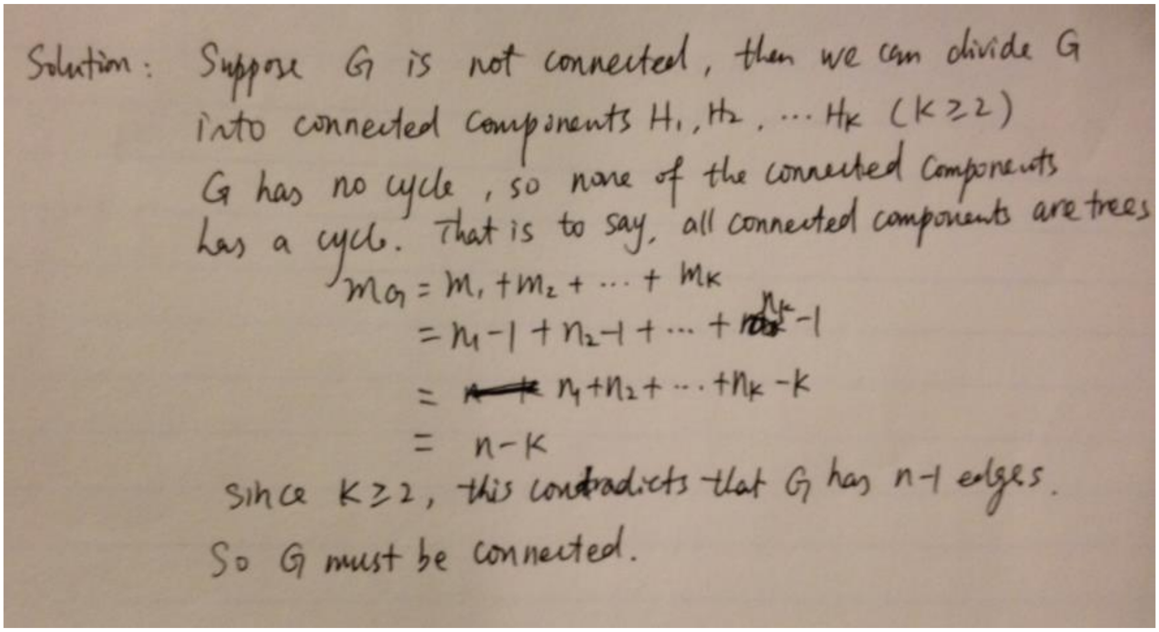
Solution:

```
Algorithm: SmallestVertexCover
   Input: A graph G whose set of vertices is denoted V and set
      of edges is denoted E
   Output: Smallest size of a vertex cover U for G

   pow ← PowerSet(V)
   minCover ←  V
   minVal ← |V|
   for each U in pow do
      isCover ← true
      //verify U is a vertex cover
      for each e in E do
         (u,v)  ← computeEndpoints(e)
         if( !(belongsTo(u,U) and !belongsTo(v,U))
            isCover ← false

      if(isCover and U.size() < minCover.size()) then
            minCover ←  U
            minVal ←  |U|
   return minVal
```

4. Show that if a graph G has |V| -1 edges and has no cycle, then G is connected.
   Hint: Assume G is disconnected, with connected components H1, H2, .. Hk. What can you say about each of these components? Do a computation to show that G must in this case have fewer than |V| - 1 edges (giving a contradiction).



Solution: Suppose G is not connected, then we can divide G into connected components $H_1, H_2, \cdots H_k$ $(k \geq 2)$
G has no cycle, so none of the connected components has a cycle. That is to say, all connected components are trees

$$\begin{aligned} m_G &= m_1 + m_2 + \cdots + m_k \\ &= n_1 - 1 + n_2 - 1 + \cdots + n_k - 1 \\ &= n_1 + n_2 + \cdots + n_k - k \\ &= n - k \end{aligned}$$

Since $k \geq 2$, this contradicts that G has $n-1$ edges.
So G must be connected.

Problem 5. Show that the VertexCover decision problem belongs to NP.

Solution. Assume a solution U of vertices is given as a solution to the VertexCover problem with input G = (V, E), k. To verify that U is correct, we must

(1) Show that $U \subseteq V$
(2) Show that U is a vertex cover – verify that each edge in E has an endpoint in U
(3) Show that U.size $\leq$ k

(1) requires O(n) (using hashtable for storing sets); (2) requires O(m) (where m is the number of edges). And (3) is (under reasonable assumptions) O(1). Therefore time to check correctness is O(n^2) and hence VertexCover is in NP.