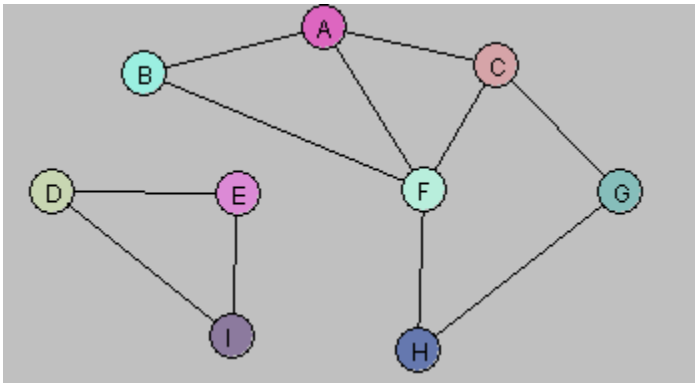
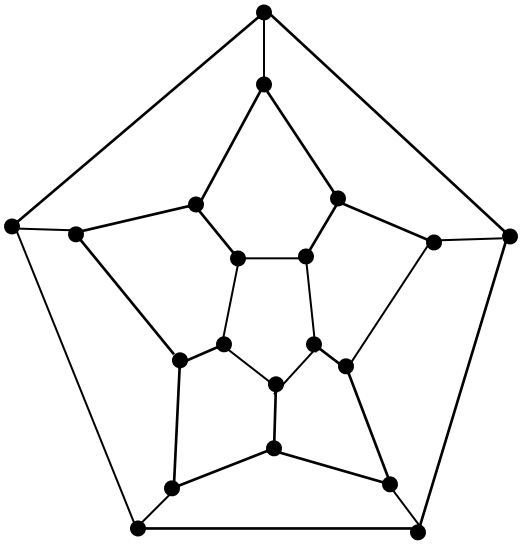
Lab 11

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



   A. Is the graph G connected? If not, what are the connected components for G?
   B. Draw a spanning tree/forest for G.
   C. Is G a Hamiltonian graph?
   D. Is there a Vertex Cover of size less than or equal to 5 for G? If so, what is the Vertex Cover?

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



3. *Vertex Covers.* Create an algorithm for computing the smallest size of a vertex cover for a graph. The input of your algorithm is a set V of vertices along with a set E of edges. Assume you have the following functions available (no need to implement these):
    - computeEndpoints(edge) – returns the vertices that are at the endpoints of the input edge
    - belongsTo(vertex, set) – returns true if the input vertex is a member of the given set

    Hint: Loop through all subsets of V. For each subset W, check to see if W is a vertex cover. Do this by looping through all edges; for each edge e, check to see if at least one of its endpoints lies in W.

4. *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?

5. 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. (You can assume G is connected.) 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.