**Algorithm: Lab11 (By Sujiv Shrestha ID:610145)**

**Problem 1.**

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?

Answer: The graph G is disconnected. The connected components for G are G1 = ({D,E,I}, {D-E,D-I,I-E}) and G2 = ({B,A,C,F,G,H},{B-A, A-C, A-F, B-F, F-C, F-H, C-G, G-H})

B. Draw a spanning tree/forest for G.

C. Is G a Hamiltonian graph?

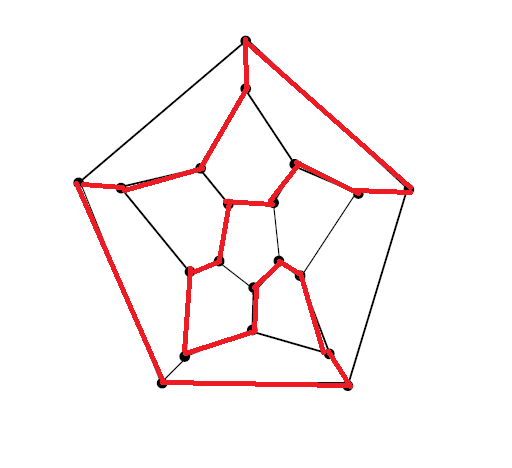
Answer: G is not a Hamiltonian graph because in Hamiltonian graph there should be atleast one cyclic path (Hamiltonian cycle) connecting all the vertices in that graph but here the graph is disconnected.

D. Is there a Vertex Cover of size less than or equal to 5 for G? If so, what is the Vertex Cover?

Answer: Yes, there is a Vertex Cover of size less than or equal to 5 for G. That Vertex Cover is V={D,E,F,G,A}

**Problem 2.**

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

****

**Problem 3.**

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.

Solution:

**Algorithm** vertexCover(V, E)

**Input** set of vertex V and set of Edges E

**Output** vertex Cover VC

VC←emptyList of vertex

**while** (¬E.isEmpty()) **do**

E1 ← E.removeFirst()

(V1,V2) ←computeEndpoints(E1)

**if**(¬belongsTo(V1, VC) **and** ¬belongsTo(V2, VC)) **then**

VC.insertLast(V2)

**return** VC