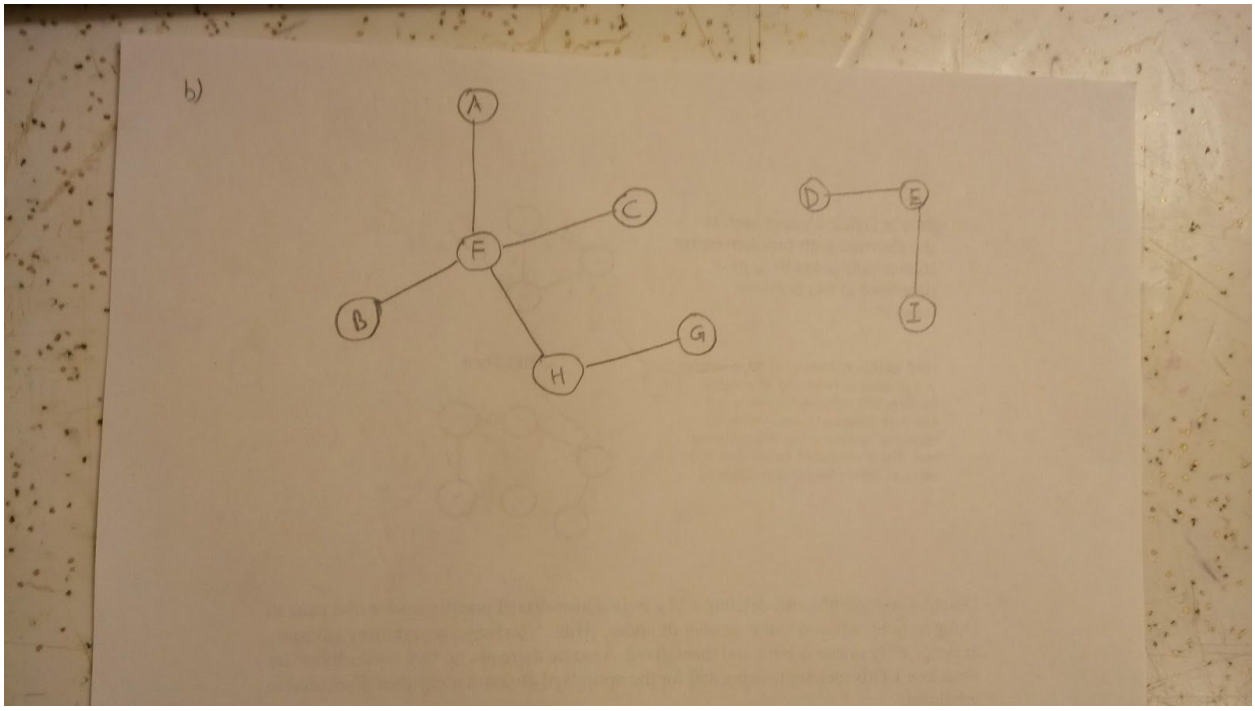
Problem 1

a) The graph is not connected. It has 2 connected components

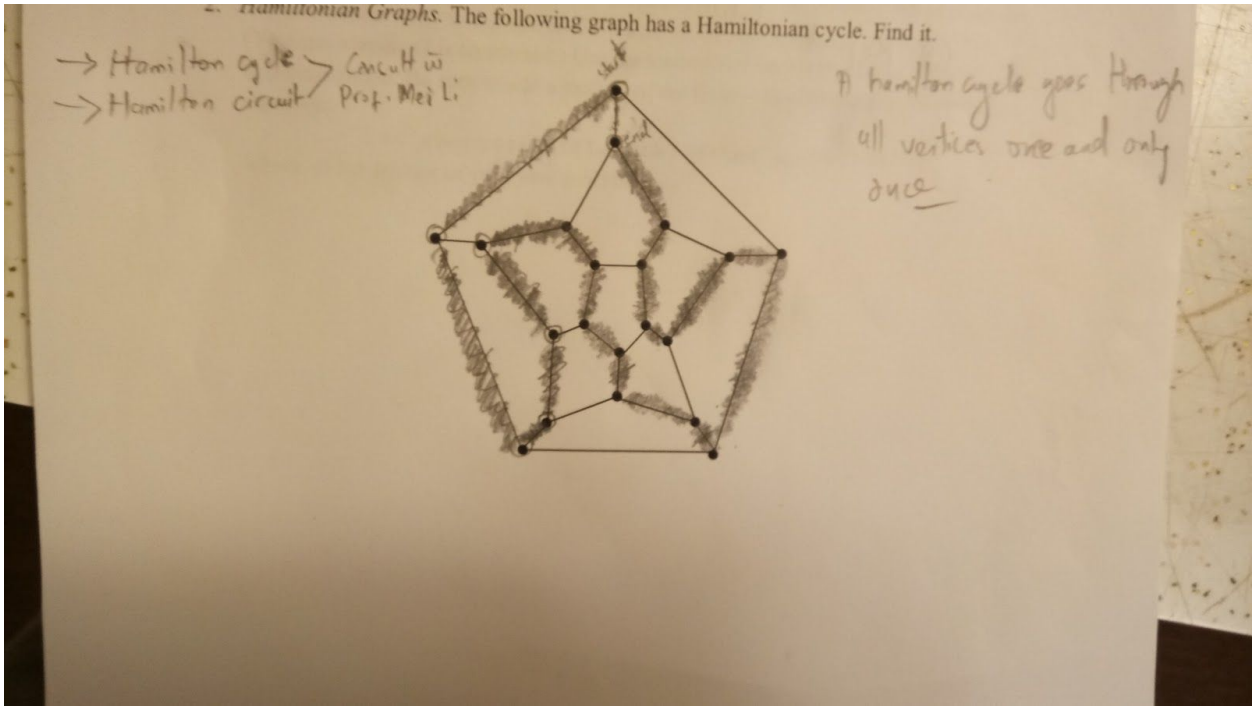
b) Spanning forest



c) The graph cannot have a hamilton cycle because it's disconnected. Note that a hamilton cycle goes through all of vertices once.

d) A vertex cover is a set of all vertices that cover all the edges . In our case of course there exists more than one vertex covers like {F,A,G,E,D}

Problem 2



Problem 3

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
```