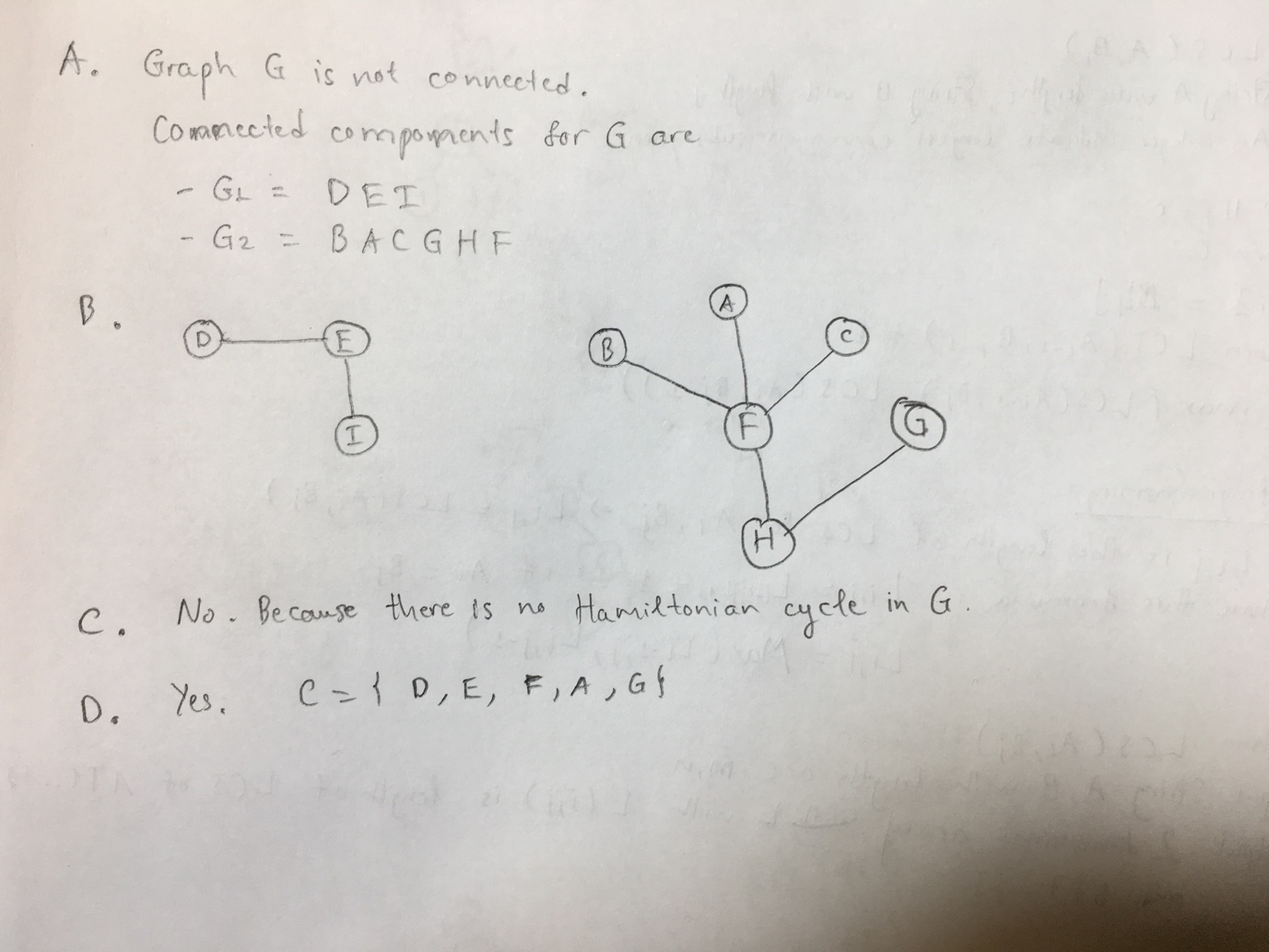
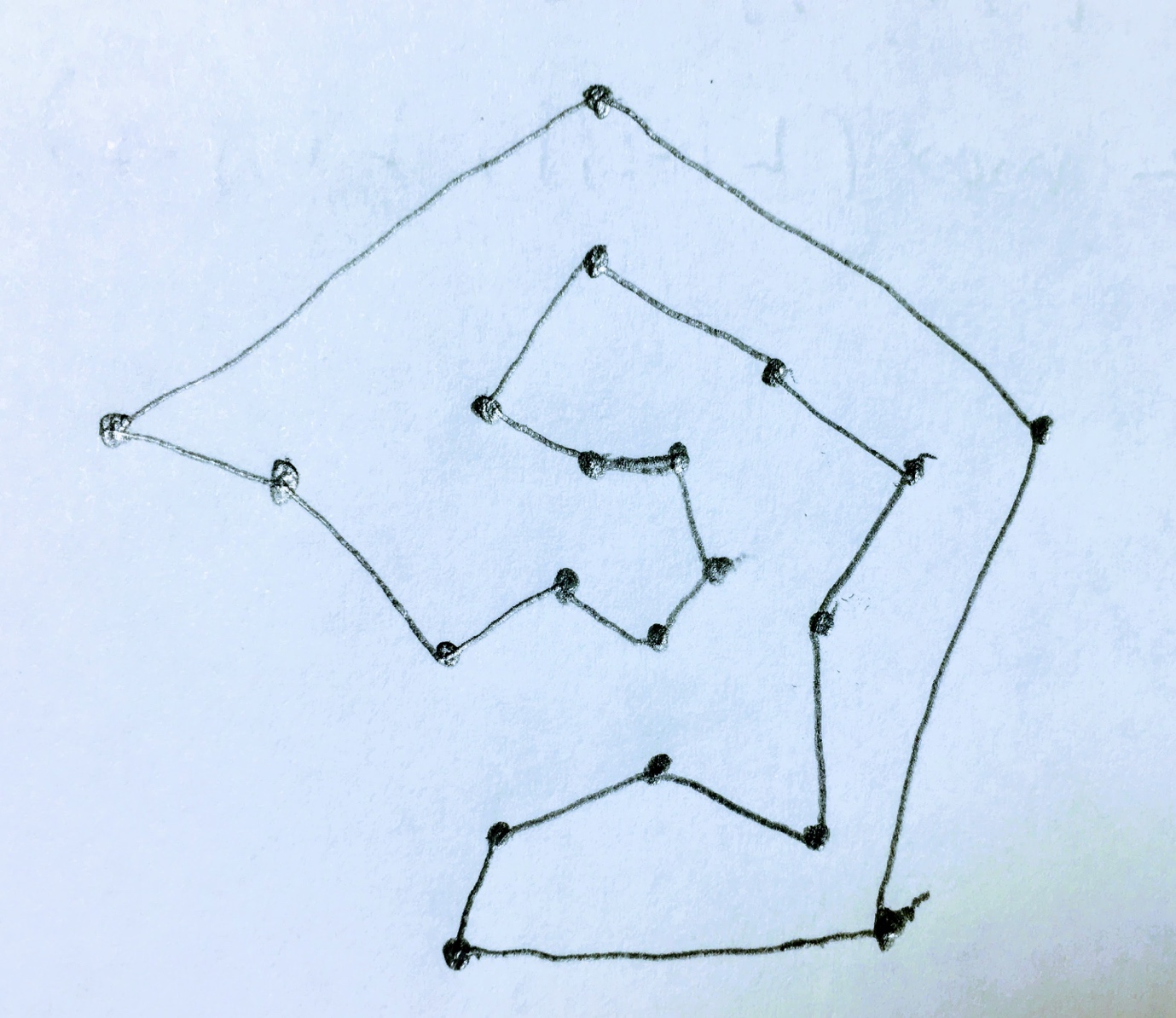
Lab 11

# Problem 1



# Problem 2



# Problem 3

Algorithm SmallestVertexCover(V, E)  
 Input      Set V of vertices, Set E of edges  
 Output Size of smallest vertex cover  
  
 S <- PowerSet(V)  
 VC <- empty set  
  
 for each s in S do  
 isVertexCover <- true  
 for each e in E do  
 (x,z) <- computeEndpoints(e)  
 if !belongsTo(s, x) && !belongsTo(s, z) then  
 isVertexCover <- false  
 break  
 if isVertexCover then  
 VC.add(s)  
  
 min = VC.get(0).size()  
 for each vc in VC do  
 if vc.size() < min then  
 min = vc.size  
  
 return min

# Problem 4

The algorithm for verify this solution is as following

Algorithm VerifyVertexCover(G(V, E),C,k)

Input A graph G with set of vertices V and set of edges E, an integer k

Output Is there a vertex cover for G having size <= k?

1. Check C ⊆ V

(2) For e E, every e has at least endpoint lies on C

(3) |C| <= k

1. Using HashMap for store every set of V => check C ⊆ V take O(n)
2. O(m) (m is number of edges, m = n \* (n - 1) / 2)
3. O(1)

T(n) = O(n2) so the algorithm VerifyVertexCover is in NP