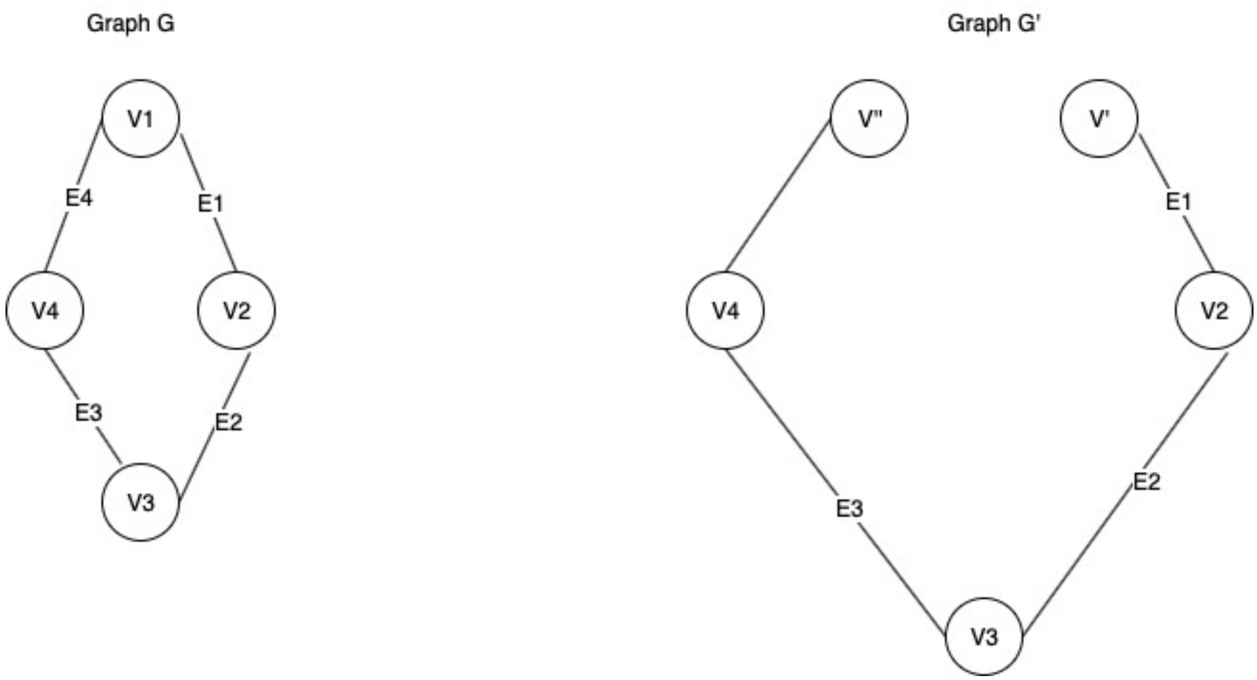
Question 1.
a)If Y is NP-complete then so is X.
No. We need to prove that all other problems in NP, or a known NP-hard problem, can reduce to X and that X is in NP.
b) If X is NP-complete then so is Y.
No. We need to show that Y is in NP.
c) If Y is NP-complete and X is in NP then X is NP-complete.
No. We need to show that all other problems in NP, or a known NP-hard problem, can reduce to X.
d) If X is NP-complete and Y is in NP then Y is NP complete.
Yes. Since X is NP-complete, it is also NP-hard, which means that all other problem in Np can reduce to X. Since X can also reduce to Y, Y is NP-hard. Therefore Y is both NP-hard and in NP, and thus is NP-complete
e) if X is in P, then Y is in P.
No. We need to prove that Y can reduce to X.
f) If Y is in P, the X is in P.
Yes. Since we can reduce X to Y, the solution to Y is also a solution to X. And since the solution can be solved in P for Y, it can be solved in P for X.

Question 2.
a) $SUBSET - SUM \ reduces \ to \ COMPOSITE$
Not possible. COMPOSITE is only NP. In order for other NP problems to map COMPOSITE, COMPOSITE must also be NP-hard.
b) If there is a polynomial algorithm for SUBSET-SUM, then there is a polynomial algorithm for COMPOSITE.
Possible. Since SUBSET-SUM is NP-complete, all other problem in NP, including COMPOSITE, can reduce SUBSET-SUM. Since SUBSET-SUM has a polynomial algorithm, COMPOSITE does too.
c) If there is a polynomial algorithm for COMPOSITE, then P=NP.
Not possible. COMPOSITE is not NP complete. Other problems in NP cannot reduce to COMPOSITE.
d) If P != NP, then no problem in NP can be solve in polynomial time.
P is a subset of NP. Since P can be reduced to NP-complete, the NP-complete problems would not be solvable in polynomial time. However, those NP problems that are not NP-Complete may still be solvable in polynomial time.

Question 3.
In order to prove that that a Hamiltonian-Path is NP-complete, we must prove that:
- HAM-PATH is in NP
- That we can reduce a known NP-complete problem to HAM-PATH, we will use HAM-CYCLE.

1. Prove that HAM-PATH is in NP:
$Given\ a\ graph\ G = \{V, E\}\ and\ a\ certificate\ , y =< e_1, e_2, \dots, e_n >, where\ e \in E, and\ the\ where\ the\ edges\ in\ the\ certificate\ are\ listed\ in\ the\ order\ they\ are\ to\ be\ selected.$
$We\ can\ verify\ whether\ this\ solution\ contains\ a\ Hamiltonian\ path.$
$This\ takes\ at\ most\ O(n), which\ is\ polynomial\ time.$
$HAM - PATH\ is\ in\ NP.$
2. Prove that HAM-CYCLE can be reduced to HAM-PATH.
$Suppose\ G\ is\ graph\ that\ contains\ a\ HAM - CYCLE, it\ is\ true\ that\ start\ and\ end\ vertex\ v\ are\ the\ same.\ Our\ reduction\ function\ will\ split\ vertex, v\ into\ two\ vertices, v'\ and\ v''\ and\ will\ have\ the\ same\ ordering\ of\ vertices\ as\ in\ G, but, instead\ starting\ and\ ending\ at\ vertex\ v, it\ will\ start\ at\ vertex\ v'\ and\ end\ at\ v'', this\ transformation\ can\ be\ done\ in\ polynomial\ time, adding\ a\ vertex\ O(1) and\ re - drawing\ the\ edges\ O(E), T(n) = O(E).$



If G is HAM-CYCLE, the transformation function will split start and end vertex, v, into two vertices, v’ and v’’. Since the ordering of nodes will be the same, the resulting graph, G’ will be a Hamiltonian Path.
If G’’ is a HAM-PATH, the transformation function will combine the start and end vertices, v’ and v’’, into one vertex v. Since the ordering nodes will be the same, the resulting graph G, will be a Hamiltonian Cycle.
$Since\ we\ have\ shown\ that\ HAM - PATH\ is\ in\ NP\ and\ NP - HARD, it\ is\ NP - complete.$
QED.

Question 4.

a.)

Pseudo-Code for 2 Color Decision Problem

```
twoColor(G, s){
  isTwoColor = True
  Let notVisited{ }be the set of vertices that have not been visited
  notVisited ← (V − {s})
  Let Q be a Queue
  Q ← ∅
  enqueue(Q, s)

  while Q is not empty{
    u = dequeue(Q)
    for each v ∈ G.Adj(u){
      if v.color == u.color{
        isTwoColor = False
        enqueue(Q, v)
      }
    }
    notVisited ← (notVisited − {u})
    if Q is empty and notVisited is not empty{
      let x be the first element in notVisited
      enqueue(Q, x)
    }
  }
  return isTwoColor
}
```

Running time:

While loop:

- Each Vertex will be visited, enqueue to Q: T(n) = O(V)
- For each vertex v, the adjacency list will be searched, in the worst case each vertex is connected to every other vetex, $O(V^2)$.
- Overall running time for while loop: T(n) = $O(V^2)$.

Searching the notVisited set for an already visited vertex: T(n) = O(n)

Removing a visited vertex from the notVisited set: T(n) = O(1)

Overall running time: T(n) = $O(V^2)$.+ O(n) + O(1) = $O(V^2)$.

b.) Prove that 4-Color is NP Complete.

In order to show that 4-Color is NP complete we need to show that it is in NP and that 3-Color can be reduced to 4-Color

1. Prove that 4-Color is in NP

Given a graph G={V,E}, and a solution, y = < (v1, c1), (v2, c2), … (vn, cn)>, where $v \in V$.

$Let\ C\ be\ the\ set\ of\ colors, c \in C, and\ |C| \leq 4$:

- We color each vertex according to the solution, this takes O(V)
- For each vertex, we check each vertex is connected to every other vetex, this will take $O(V^2)$.
- If each parent and child have different colors, then the verification passes.

Our verification algorithm $O(V^2)$, therefore the problem is in NP.

2. Prove that 3-Color can be reduced to 4-Color.

Our reduction algorithm will take a Graph, G, and produce a new graph G' which is a 4-Color graph if and only if G is a 3-Color Graph.

G' is created by adding new vertex y, which is a color not in G. Every other vertex is then connected to y. Adding a new vertex is O(1), creating an edge from every other vertex to y is O(V). The overall running time of this reduction O(V) which is polynomial.

If G is a 3-Color graph, applying the reduction algorithm described above will result in a graph G' that is 4-Color. This is because the original vertices are copied over and vertex y is the only node with a new color. G' is a 4-Color Graph if and only if G is a 3-Color Graph.

Conversely, if G' is a 4-Color graph, then y is the only vertex of its color, removing this vertex and every edge connected to it, will result a 3-Color Graph. G is a 3-Color Graph if and only if G' is a 4-Color Graph.

Since we have shown that 4-Color is in NP and NP-hard, it NP-Complete. QED.