Homework 6

Sahil Palnitkar
CSCE 411

34.2-10)
We can prove this by proving the contrapositive. That is, if P = NP, then NP = CoNP.
Assume P = NP.
If P = NP, then as P is closed under complement, NP is also closed under complement.
CoNP is the complement of all problems in NP. Therefore CoNP = NP.

Since the contrapositive is true, the original statement will be true. Therefore, if NP != CoNP, then P != NP.

34.5-1)
First to show that subgraph isomorphism is in NP. Let G be the subgraph of G2. We know the mapping between vertices of G1 and G. We can verify in **polynomia**l time that for every edge in G1, there is an edge present in G. That means that G1 is isomorphic to G. This means that this problem is **in NP.**

To show that the problem is NP-Hard, we can reduce to the CLIQUE problem with input (G, k). The clique problem is true if the graph G contains a clique of size k (subgraph of G).
Let G1 be a compete graph of k vertices and G2 be G, where G1 and G2 and inputs to the subgraph isomorphism problem. Then k $\leq$ |G2|. If k > |G2|, a clique of size k cannot be a subgraph of G. Time taken to make G1 is O(n^2) as the no. of edges in a graph of size k = ${}^{k}C_{2}$. G will have a clique of size k iff, G1 is a subgraph of G2. As every graph is isomorphic to itself, if G1 is a subgraph of G2, the result of the subgraph isomorphism is true. Therefore the Clique problem can be reduced to subgraph isomorphism in **polynomial** time. Therefore subgraph isomorphism is **NP-hard.**

Since the subgraph isomorphism problem is in NP and is NP-Hard, it is **NP-Complete**.

34-1)
a) The certificate will be a set of vertices of at least size k. We can reduce the CLIQUE problem to have input <G,k> to show NP-completeness. We can use the complement G'. G will have a CLIQUE of size k iff G' has an independent set of size k. This proves that Clique $\leq_p$ Independent Set. Since Clique is NP-complete, the Independent Set problem is also NP-complete.
b) We can run the subroutine on <G,k> for k = 1,…n -> (|V|) to find the largest k = k0, such that the subroutine will return true. We can pick a vertex V in G and denote the subgraph of G constructed by removing V and its induced edges by G(V'). We can run the subroutine on (G(V'),k0). If the subroutine returns true, we

can recursively find an independent set of size k0-1 in G(V') -> (|E|). This runs in polynomial time.