Question 1 Solution

1. To show that SUBGRAPH - ISO $\in NP$, we need to show that it has a polynomial time verifier.

The certificate c, will be the function f and the subgraph G.

$V =$ " On input $\langle\langle G', H\rangle, c\rangle$:

1. Let $c = (G, f)$. First check that $|G| = |H|$. If not, reject
2. Let P be the $\binom{n}{2}$ collection of pairs of vertices in G where $|G| = n$. For each $p \in P$:
3. Check that g_1 and g_2 s.t $p = (g_1, g_2)$ are adjacent in G iff $P_h = (f(g_1), f(g_2))$ are adjacent in H.
4. If true for all $p \in P$ in step 3, then accept, otherwise reject.

"

All that remains to be shown is V is a polynomial time verifier.

Step 1: runs in $O(n)$ since you just need to count the vertices in G and H

Step 2: This loop will take $O(n^2)$ time since there are $\binom{n}{2}$ pairs.

Step 3: Each iteration of the loop is $O(1)$

Step 4: $O(n^2)$, Simply check each result returned by loop in step 2.

Thus the verifier runs in polynomial time.

Question 1 : Solution

We also note that V is indeed a valid verifier. If G is isomorphic to H, step 3 checks every possible pairs of vertices in G and checks that g_1 and g_2 are adjacent in G iff $f(g_1)$, $f(g_2)$ are adjacent in H. V would then accept.

Similarly if G is not isomorphic to H, then $\exists$ a pair of vertices (g_1, g_2) s.t (g_1, g_2) are incident iff $(f(g_1), f(g_2))$ are incident is false. Step 3 would detect this and V would reject.

Question 2 : Solution

To show that SUBGRAPH-ISO is NP hard, we need to show that $L' \leq_p L$ for some other NP-hard language L'.

We will show that CLIQUE $\leq_p$ SUBGRAPH-ISO. Since we know CLIQUE is NP Hard, SUBGRAPH-ISO would be NP Hard as well.

Define M to be: "On input $\langle G', K \rangle$:

1. Construct a graph H with K vertices, each of which is adjacent to each other (so that H is a K-clique)
2. Output $\langle G', H \rangle$.

First, we show if $x \in$ CLIQUE then $f(x) \in$ SUBGRAPH-ISO. If $x \in$ CLIQUE that means G' has a subgraph, G, that is a CLIQUE of size K. By construction, H is a K-clique.

Clearly then, G must be isomorphic to H since they are both K-cliques and all vertices are incident to each other.

Question 2: Solution.

Next, we show if $x \notin$ CLIQUE then $f(x) \notin$ SUBGRAPH-ISO

Since $x \notin$ CLIQUE, then we know that $f(x) = \langle G', H \rangle$ will be such that H is a k-clique by construction, but G' does not contain a clique of size k.

If G' does not contain a clique of size k, it is impossible for there to exist some subgraph G that is isomorphic to H.

If there did exist such a G, by definition of G being isomorphic to H, G must be k vertices all incident to each other implying G is a k-clique, which is a contradiction. Thus, $f(x) \notin$ SUBGRAPH-ISO.

Now, all that remains to be shown is $f(x)$ runs in polynomial time. Clearly this is so, because step 1 is simply construct a k-clique (independent from G'), and step 2 is just outputting $\langle G', H \rangle$, both steps run in polynomial time so $f(x)$ is polynomial.

Question 3

a) First, notice that we are constrained by the fact that $x_u + x_v \geq 1$ for every edge $\{u, v\} \in E$.

Thus if $C = \{v \in V : x_v \geq 1/2\}$ then C must be a vertex cover.

Suppose C was not a vertex cover. That means $\exists$ an edge $\{u, v\} \in E$ s.t $u, v \notin C$.

But if $u, v \notin C \Rightarrow x_u, x_v < 1/2$
But this implies $x_u + x_v < 1$, which is a contradiction of our original constraint.

b) Let C' be the optimal vertex cover. Clearly $|C'| \geq \frac{|V|}{2}$.

However, notice that $|C| \leq |V|$. In fact $C = V$ if the polynomial time algorithm simply sets all the values for $x_v = 1/2$. Thus, we have

$$\frac{|V|}{2} \leq |C'| \leq |C| \leq |V|$$

So C is a 2-approximation of a minimum weight vertex cover of G.

an S ... A verifier for a language A is an algorithm V, where "certificate"
$A = \{w \mid V$ accepts $\langle w, c\rangle$ for some string $c\}$
We measure the time of a verifier only in terms of length of w.
polynomial time verifier runs in polynomial time in length of w.
language A is polynomially verifiable if it has a polynomial time verifier.

At any given point, you can ... branches.

NP Hard: If L is NP-hard, that means for every other language A, $A \le_p L$. (It suffices to show that $L' \le_p L$ for some other NP-hard.
NP Complete: L is NP complete if $L \in NP$ & $L \in$ NP Hard

NP: is the class of languages that have polynomial time verifiers.
coNP: class of languages whose complements are in NP (coNP is not the complement of NP)

Goal: Show a problem is in NP / NP Complete

CLIQUE: $\{\langle G, k\rangle \mid G$ is an undirected graph with a k-clique$\}$

SUBSET-SUM: $\{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$ for some $\{y_1, \ldots, y_\ell\} \subseteq \{x_1, \ldots, x_k\}$, we $\sum y_i = t\}$
is NP Complete

Proof idea: c = the clique

"On input $\langle\langle G, k\rangle, c\rangle$:
1. Test whether c is a subgraph with k nodes in G.
2. Test whether G contains all edges connecting nodes in c
3. If both pass, accept; otherwise, reject."

Definition 7.29: Language A is polynomial time mapping reducible to language B, written $A \le_p B$, if a polynomial time computable function $f: \Sigma^* \to \Sigma^*$ exists, where for every w $w \in A \iff f(w) \in B$

Problem Set 14.1:
Example of Polynomial time reduction from INDSET to MFE

INDSET $= \{\langle G, k\rangle : G = (V, E)$ is an undirected graph and $1 \le k \le |V|$ is an integer s.t G has an independent set of size $k\}$.
independent set of vertices: No two vertices are adjacent.

MFE $= \{\langle G, C, k\rangle : G = (V, E)$ undirected graph, C is a collection of pairs of edges in G, $1 \le k \le |E|$ is an integer s.t G has a matching of size k which respects $C\}$
matching M respects ... if it contains at most one edge from each pair in C.

Show INDSET $\le_p$ MFE
INDSET: Find a group of vertices S, $|S| = k$ s.t for any $e_{v_1, v_2} \in E$ only v_1 or v_2 is in S. This is similar to how for any given pair of edges $c \in C$, only 1 can be included in M.
For every $v \in V$ of INDSET, create an

Problem Set 14.4: Show CLIQUE $\le_p$ DENSE SUBGRAPH
$(G, k) \to (G, k, 1)$
WTS: $x \notin$ CLIQUE $\Rightarrow f(x) \notin$ DENSE-SUBGRAPH

Problem Set 14.2:
Let $[n]$ denote the set of all integers from 1 to n (inclusive). The SET-COVER problem is:
$\{\langle n, m, C, k\rangle : C = \{S_1, \ldots, S_m\} \subseteq 2^{[n]}$ and $\exists I \subseteq [m]$ s.t $|I| \le k$ and $\bigcup_{i \in I} S_i = [n]\}$
Intuitively, there exists a collection of subsets of size $|I|$ such that the union of these subsets equals $[n]$

VERTEX-COVER: $\{\langle G, k\rangle \mid G$ is an undirected graph that has a k-node vertex cover$\}$
Show VERTEX-COVER $\le_p$ SET-COVER

$M_{vc\text{-}2\text{-}sc}$ "On input $\langle G, k\rangle$, where $G = (V, E)$ is an undirected graph and k is an integer $1 \le k \le |V|$, construct $\langle n, m, C, k'\rangle$ as follows:

$O(1)$ 1. Let $n = |E|$. Let $m = |V|$. Let $k' = k$. Initialize $C = \emptyset$
$O(n)$ 2. Label each edge E_i in G with a unique integer 1 to n.
$O(n)$ 3. For each vertex V_j in V:
$O(|V||E|)$ 4. - Create a new subset, S_j and add the integers assigned to edges incident on V_j to S_j.
$O(n)$ 5. - Add S_j to C
$O(n)$ 6. Output $\langle n, m, C, k'\rangle$
"

WTS: $x \in$ VERTEX COVER $\Rightarrow f(x) \in$ SET COVER.
Since $x \in$ VERTEX COVER, $\exists V_c$ of size k in G. Basically in Step 4, we create a new subset S_j based on edges incident to $v \in V_c$.
Since V_c is a vertex cover, we reach all edges, thus, $f(x) \in$ SET COVER
WTS: $x \notin$ VERTEX COVER $\Rightarrow f(x) \notin$ SET COVER
Since $x \notin$ VERTEX COVER, it's impossible to select k subset markers include all integers.
Thus, it's impossible for $f(x) \in$ SET COVER
WTS $M_{vc\text{-}2\text{-}sc}$ runs in polynomial time.
Each step in $M_{vc\text{-}2\text{-}sc}$ runs in polynomial time

Problem Set 14.3:
Show 3CNF-SAT $\le_p$ 4CNF-SAT
3CNF-SAT: all clauses have 3 literals, restricted by $\wedge$.
Eg: $(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_5 \vee x_6) \wedge (x_3 \vee \bar{x}_6 \vee x_4) \wedge (x_4 \vee x_5 \vee x_6)$
To show 3CNF-SAT $\le_p$ 4CNF-SAT, WTS: Convert clause with 3 literals to 4 literals.

$(l_1 \vee l_2 \vee l_3)$
$\downarrow$
$(l_1 \vee l_2 \vee l_3 \vee l_4) \wedge (l_1 \vee l_2 \vee l_3 \vee \bar{l}_4)$
if $l_1 \vee l_2 \vee l_3$ is all true
$l_1 \vee l_2 \ldots$ false $\to$ false

ractice Exam 5 #2.
H be a subgraph of an undirected graph G. Show
OT-MAX-CLIQUE is NP-hard.

do this we can show CLIQUE $\leq_p$ NOT-MAX-CLIQUE
nce we know CLIQUE is NP-Hard, this would imply NOT-MAX-CLIQUE is NP-Hard.

fine M to be: "On input <G, K>.

1. If K = 1 & G has at least one vertex, output <G', H> where G' is a graph consisting of two vertices conne by an edge and H consists of just one vertex of G'
2. Otherwise if K = 1 & G has no vertices or if K > |V|, output <G, H> where H is a graph wit one vertex
3. Otherwise, construct a graph H with K-1 vertices, each of which is adjacent to each other (so th H is a (K-1)-clique).
4. Let G' be the graph consisting of the disjoint union of G and H (so that no edges connect a vertex in G and a vertex in H.
5. Output <G', H>

(H is a clique, but H is a clique of size K-1, but we know G has clique size |K|

x ∈ CLIQUE, then f(x) ∈ NOT-MAX-CLIQUE since
x ∉ CLIQUE, then f(x) ∉ NOT-MAX-CLIQUE since H is the largest clique.
ch step also takes polynomial time, so f is indeed a polynomial reduction.

oblem Set 13.5 c

w that BIPARTITE = {<G> : G is an undirected bipartite Graph} ∈ NP ∩ coNP
ow BIPARTITE ∈ NP
is (L, R)
On input <<G>, c>,
1. Check c, L and R make up G
2. There are no edges between L and R

how BIPARTITE ∈ coNP
c is an odd cycle.