#1

$x_1 = 1 \quad n_1 = 2 \quad m_1 = 15$
$x_2 = 4 \quad n_2 = 5 \quad m_2 = 6$
$x_3 = 2 \quad n_3 = 3 \quad m_3 = 10$

$n = n_1 n_2 n_3 = 30$

$\therefore \quad 15^{-1} = 1 \pmod 2$
$6^{-1} = 1 \pmod 5$
$10^{-1} = 1 \pmod 3$

$\therefore \quad m_1^{-1} = 1, \; m_2^{-1} = 1, \; m_3^{-1} = 1$

$\therefore \quad C_1 = 15 \times (1 \bmod 2) = 15$
$C_2 = 6 \times (1 \bmod 5) = 6$
$C_3 = 10 \times (1 \bmod 3) = 10$

$$x = \sum_{i=1}^{3} x_i C_i$$

$= 1 \times 15 + 4 \times 6 + 2 \times 10 \pmod{30}$
$= 15 + 24 + 20 \pmod{30}$
$= 59 \pmod{30}$
$= 29 \pmod{30}$

Hence, the solution of the given equations are of the form $29 + 30 \times (n)$ for $n \geq 0$

3.

1) give the RSA transformation P and S.

$P(M) = M^e \pmod n$, $S(c) = c^d \pmod n$ $0 \le M < n$

e is relatively prime to $\phi(n) = (p-1)(q-1)$

Compute the multiplicative inverse d of e mod $\phi(n)$

2) Prove the transformations are inverse of each other.

Proof:

$P(S(M)) = S(P(M)) = M^{ed} \pmod n$. Since e and d are multiplicative inverses modulo $\phi(n) = (p-1)(q-1)$,

$ed = 1 + k(p-1)(q-1)$

for some k. If $M = 0 \pmod n$, then $M^{ed} \equiv 0 \pmod n$. So we are done.

If M is not congruent to $0 \pmod p$, we have

$$M^{ed} \equiv M(M^{p-1})^{k(q-1)} \pmod p$$

$$= M(1)^{k(q-1)} \pmod p$$

$$= M \pmod p$$

and a similiar result holds mod q. By the Chinese remainder theorem, this implies $M^{ed} \equiv M \pmod n$.

4.

Suppose we are given a graph $G = (V, E)$ and an integer k. The certificate we choose is the vertex cover $V' \subseteq V$ itself. The verification algorithm affirms that $|V'| = k$, and then it checks, for each edge $\langle u, v \rangle \in E$, that $u \in V'$ or $v \in V'$. This verification can be performed in polynomial time.

Why polynomial time verification?
There are $|V|$ vertices, $v_1, v_2 \cdots v_{|V|}$, checking each v_i has an edge to the vertex cover V', which has the size at most k, can be done in quadratic time.

5 Suppose I slightly tweaked the problem on HW.

Suppose φ is an instance of SAT with n variables.

Consider the following formula:

$F := (\varphi \wedge a) \vee (\varphi \wedge b) \vee (\neg\varphi \wedge a_1 \wedge a_2 \wedge \cdots \wedge a_{2n})$

Here

If φ is satisfiable, let v be a minimal assignment s.t. $\bar{v}(\varphi) = T$. Then if either

$v \cup \{v(a) = T, v(b) = F\}$

$v \cup \{v(a) = F, v(b) = T\}$ is a minimal assignment making $\bar{v}(F) = T$. Notice the minimal assignment v' with $\bar{v}'(\neg\varphi \wedge a_1 \wedge \cdots \wedge a_{2n}) = T$ must set at least $2n$ var's true w/c is more than $\bar{v}$ does.

On the other hand, if φ is unsatisfiable, the only way to make F true is to make

$(\neg\varphi \wedge a_1 \wedge \cdots \wedge a_{2n})$ true

The assignment w/c makes each var in φ false and each a_i true will be the unique minimal assignment.

F can be computed from φ in polynomial time so this gives the reduction.

6. Proove TAUTOLOGY is co-NP-complete.

Let ϕ be a boolean formula.

$TAUTOLOGY = \{\phi \mid \phi \text{ is always true}\}$

The language is coNP if its complement is in NP.

$\overline{TAUTOLOGY} = \{\phi \mid \phi \text{ is false}\}$.

An algorithm can be designed that guesses the truth assignment and checks that it is unsatisfiable and evaluating a formula can be done in polynomial time.

So, $\overline{TAUTOLOGY} \in NP$.

Show that $\overline{TAUTOLOGY}$ is NP-HARD.

$SAT \leq_p \overline{TAUTOLOGY}$

If ϕ is satisfiable, then it is not always false.

$L \in coNP$.

We'd like to show $L \leq_p TAUT \Rightarrow \exists f$ $x \in L$ iff $f(x) \in TAUT$.

We showed $\exists g$ $x \in SAT$ iff $g(x) \in \overline{TAUT}$.

Consider $\overline{L}$, we know $\overline{L} \in NP$.

$\therefore \exists f'$ s.t.

$x \in \overline{L}$ iff $f'(x) \in SAT$ iff $g(f(x)) \in \overline{TAUT}$

$\Rightarrow x \in L$ iff $g(f(x)) \in TAUT$.

(Q.7) Prove the TSP where the cost function satisfies the triangle inequality has a 2 approximation scheme

Consider graph $G = \langle V, E \rangle$.
A minimal spanning tree algorithm runs in $O(|V|^2)$ time. The remaining steps take $O(|G|)$ time.

Let H^* denote the optimal tour of vertices. We obtain a spanning tree from any tour by deleting an edge, we have, $C(T) \leq C(H^*)$ where T is a minimal spanning tree.

A full-walk F of T lists the vertices when they are first visited and also whenever they are returned to after a visit to a subtree.

$$\therefore \quad C(F) = 2 \cdot C(T) \leq 2 \cdot C(H^*).$$

On the other hand, the H returned by the algorithm satisfies,

(8) Give an $8/7$ - approximation scheme for MAX-3-CNF.

Randomly set each variable in the MAX-3-CNF instance to 1 or 0 with probability of $1/2$ in each case. This is a "randomized" $8/7$-approximation algorithm.

Let $Y_i = I\{\text{clause } i \text{ is satisfied}\}$

$Y_i = 1$ as long as at least one of the literals in the i^{th} clause is set to 1.

So a clause is not satisfied only if all 3 of its literals are set to 0.

So $Pr\{\text{clause } i \text{ is not satisfied}\} = (1/2)^3 = 1/8$

Thus: $Pr\{\text{clause } i \text{ is satisfied}\} = 1 - 1/8 = 7/8$

$\Rightarrow E[Y_i] = \frac{7}{8}$

Let $Y = Y_1 + Y_2 + \cdots + Y_m$

By linearity of expectations

$$E[Y] = E\left[\sum_{i=1}^{m} Y_i\right] = \sum_{i=1}^{m} E[Y_i] = \sum_{i=1}^{m} 7/8 = \frac{7m}{8}$$

m is the upper bound on the # of satisfied clauses, hence the approximation ratio is at most $\frac{m}{(\frac{7m}{8})} = \frac{8}{7}$

(1) Carmichael # : odd Composite ## n which satisfies fermat's little theorem

$$a^{n-1} \equiv 1 \pmod{n}$$

Pseudo Prime

Composite # that passes a test on sequence of test that fails for most composite numbers

A number is pseudoprime for a, if it is Composite but $a^{n-1} \equiv 1 \pmod{n}$

(2) When we prove that n cannot be a prime power.

It is also used in case when

$X = \langle \ldots, d, 1, \ldots, 1 \rangle$ where $d \neq \pm 1$

and d is a nontrivial square root of 1.

Q10. To prove : $\{x \mid x \text{ is a PRIME}\}$ is in co-NP

co-NP is defined as set of languages L such that $\bar{L} \in NP$

COMPOSITE is NP as one can decide compositeness by non-deterministically guessing a factor

$\therefore$ COMPOSITE $\in$ NP

$\Rightarrow$ $\overline{\text{PRIME}} \in NP$

$\therefore$ PRIME $\in$ co-NP

Hence $\{x \mid x \text{ is a PRIME}\}$ is in co-NP