Name: SOLUTIONS

DO PROBLEM 1 and *TWO* OF THE REMAINING THREE PROBLEMS

(IF YOU DO ALL PROBLEMS ONLY 1-3 WILL BE GRADED)

Problem 1: True, False or Open. ("Open" means "nobody knows".)
Correct =1, incorrect = -1, no answer = 0.

T			If $L \in \text{P}$ then $L \in \text{NP}$ and $\bar{L} \in \text{NP}$ (Recall that $\bar{L}$ is the complement of L.)
		O	If $L \in \text{NP}$ and $\bar{L} \in \text{NP}$ then $L \in \text{P}$
T			If $L \in \text{NP}$ then $L^* \in \text{NP}$ (Recall that $L^* = \cup_{k=0}^{\infty}\{w_1 w_2 \cdots w_k : w_1, w_2, \ldots, w_k \in L\}$.)
	F		If $L^* \in \text{NP}$ then $L \in \text{NP}$
T			If A is NP-complete and $B \in \text{NP}$ and $A \leq_P B$ then B is NP-complete
		O	If A is NP-complete and $B \in \text{NP}$ and $B \leq_P A$ then B is NP-complete
T			If $B \in \text{P}$ and $A \leq_P B$ then $A \in \text{P}$
		O	There exists a polynomial-time algorithm for SAT
T			Every problem in NP is decidable
		O	There exists a regular language that is NP-complete
T			SPATH $\leq_P$ SAT (Recall that SPATH $= \{\langle G, s, t, k\rangle$: graph G has a path from s to t of length at most $k\}$).
T			$\text{E}_{\text{DFA}} \in \text{P}$ (Recall that $\text{E}_{DFA} = \{\langle D\rangle : D$ is a DFA and $L(D) = \emptyset\}$.)

Problem 2: Recall that CLIQUE $= \{\langle G, k\rangle : G$ has a clique of size $k\}$, where a clique C is a set of vertices such that for every pair $u, v \in C$, edge (u, v) is in G.

Describe a polynomial-time reduction Φ from CLIQUE to SAT. If parts of your reduction are similar to parts of reductions we've seen before, you may point out the similarity and sketch those parts rather than giving all the details. On the other hand, make sure your reduction is completely specified.

Sketch a proof that your reduction is correct. In particular, describe how, given any CLIQUE of size k in G, you can find a satisfying assignment to your formula $\Phi(G, k)$, and, conversely, how, given any satisfying assignment to $\Phi(G, k)$, you can find a CLIQUE of size k.

Given a graph G and an integer k, construct a CNF formula Φ as follows.

For each vertex v, introduce a variable X_v (the intended meaning is that X_v is true if v is in the clique).

For each pair of vertices u, v in G having no edge (u, v), add a clause $\overline{X_v} \vee \overline{X_u}$ to Φ.

At this point, the satisfying assignments to Φ correspond to cliques in G. Next we add variables and clauses to make sure that any satisfying assignment corresponds to a clique of size at least k.

Order the vertices as $v_1, v_2, \ldots, v_n$.

For each $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, k$, introduce a variable Y_{ij}. The intended meaning is: Y_{ij} is true if there are at least j true variables in the set $X_{v_1}, X_{v_2}, \ldots, X_{v_i}$. Add clauses to enforce this just as we did for the reductions from VERTEX-COVER and DOM-SET to SAT in class.

Namely, add clauses enforcing $Y_{i0} =$ true for each i, $Y_{0j} =$ false for each $j > 0$, and $Y_{ij} = Y_{i-1,j} \vee (Y_{i-1,j-1} \wedge X_{v_i})$.

Finally, add a clause containing just the variable Y_{nk}.

That's the reduction. Clearly it can be computed in polynomial time.

Next we verify it's correct.

($\Rightarrow$) Suppose G has a clique C of size k. Assign true to the variables X_v such that $v \in C$, and false to the other variables. Assign true to those Y_{ij}'s such that there are at least j true variables in $\{X_{v_1}, X_{v_2}, \ldots, X_{v_i}\}$, assign false to the other Y_{ij}'s. This gives a satisfying assignment for Φ.

($\Leftarrow$) Suppose Φ has a satisfying assignment. Let $C = \{v : X_v$ is assigned "true" $\}$. Then C is a clique because for any pair of vertices u, v in C, there cannot be a clause $\overline{X_v} \vee \overline{X_u}$ in Φ (because such a clause would not be satisfied), so there must be an edge (u, v) in G.

C has size k or more because Y_{nk} must be true.

Problem 3:

Prove one of the following two problems is NP-complete. You may use the fact that SUBSET-SUM is NP-complete. If you give a reduction, *prove that your reduction is correct.*

SUBSET-SUM-TO-HALF:

Instance: n numbers $x_1, x_2, \ldots, x_n$
Query: Is there a subset $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} x_i \;=\; \sum_{i=1..n} x_i/2$?

Clearly the problem is in NP: a verifier, given the instance and the subset S', can verify the condition in polynomial time. To complete the proof that the problem is NP-complete, we show the problem is NP-hard by reducing SUBSET-SUM to SUBSET-SUM-TO-HALF as follows.

Given a possible instance $y_1, y_2, \ldots, y_m$ and T of SUBSET-SUM:

1. *Compute y_{m+1} such that $T + y_{m+1} = \sum_{i=1..m+1} y_i/2$.*
   (Specifically take $y_{m+1} = \sum_{i=1..m} y_i - 2T$.)
2. *Output $y_1, y_2, \ldots, y_m, y_{m+1}$.*

Clearly the reduction can be done in polynomial time.

($\Rightarrow$) Suppose $(y_1, y_2, \ldots, y_m), T \in$ SUBSET-SUM. Then whatever subset sums to T, adding y_{m+1} to the subset gives a subset of $(y_1, y_2, \ldots, y_{m+1})$ summing to $\sum_{i=1..m+1} y_i/2$, so $(y_1, y_2, \ldots, y_{m+1}) \in$ SUBSET-SUM-HALF.

($\Leftarrow$) Suppose $(y_1, y_2, \ldots, y_{m+1}) \in$ SUBSET-SUM-HALF. Then there is a subset that sums to $\sum_{i=1..m+1} y_i/2$. Both this subset and its complement sum to the same thing, so there is a subset containing y_{m+1} and summing to $\sum_{i=1..m+1} y_i/2$. Taking y_{m+1} from this subset gives a subset of $(y_1, y_2, \ldots, y_m)$ summing to T. (This is true by the choice of y_{m+1}.) So $(y_1, y_2, \ldots, y_m), T \in$ SUBSET-SUM.

HALF-SUBSET-SUM:

Instance: n (not necessarily distinct) numbers $x_1, x_2, \ldots, x_n$ and a target number T
Query: Is there a subset $S' \subseteq \{1, 2, \ldots, n\}$ $\underline{\text{containing } n/2 \text{ elements}}$ such that $\sum_{i \in S'} x_i \;=\; T$?

The problem clearly has a polynomial-time verifier, so is in NP. To finish we show the problem is NP-hard by reducing SUBSET-SUM to it as follows:

Given a possible instance $y_1, y_2, \ldots, y_m$ and T of SUBSET-SUM:

1. *Let $y_i = 0$ for $i = m+1, m+2, \ldots, 2m$.*
2. *Output $y_1, y_2, \ldots, y_m, y_{m+1}, \ldots, y_{2m}$ and T*

Clearly the reduction can be done in polynomial time.

($\Rightarrow$) Suppose $(y_1, y_2, \ldots, y_m), T \in$ SUBSET-SUM. Then there are, say, some t of the y_i's that sum to T. Adding $m - t$ of the zero-valued y_i's gives m of the numbers in $y_1, y_2, \ldots, y_{2m}$ that together sum to T. So $(y_1, y_2, \ldots, y_{2m}), T \in$ HALF-SUBSET-SUM.

($\Leftarrow$) Suppose $(y_1, y_2, \ldots, y_{2m}), T \in$ HALF-SUBSET-SUM. Then some m of the above numbers sum to T. Removing the any y_i's with $i > m$ from the subset, we that some sub-collection of $\{y_1, y_2, \ldots, y_m\}$ sums to T. So $(y_1, y_2, \ldots, y_m), T \in$ SUBSET-SUM.

Problem 4: Recall that

VERTEX-COVER = $\{\langle G, k\rangle$: graph G has a vertex cover of size $k\}$.
DOM-SET = $\{\langle G', k'\rangle$: graph G' has a dominating set of size $k'\}$.

Describe a reduction from VERTEX-COVER to DOM-SET. You can use the following almost-finished reduction as a starting point, or you may give some other reduction if you prefer. In either case, prove your reduction is correct.

Given G and k, construct graph G' and integer k' as follows:

0. Add two new vertices x and y to G' with edge (x, y).
1. For each vertex v in G, make a vertex v in G' with an edge from v to x.
2. For each edge (u, v) in G, make $k+2$ vertices $uv_1, uv_2, \ldots, uv_{k+1}$ in G' and, from each of these "edge" vertices uv_i, add two edges (u, uv_i) and (v, uv_i) to u and v.
3. Set $k' =$???.
4. Output graph G' and integer k'.

To complete the reduction, line 3 should set $k' = k+1$.

Clearly the reduction is in polynomial time.

To finish we prove that $(G, k) \in$ *VERTEX-COVER* $\Leftrightarrow$ $(G', k+1) \in$ *DOM-SET.*

($\Rightarrow$*) Suppose* G *has a vertex cover* C *of size* k*. Then it is easy to verify that* $C \cup \{x\}$ *is a dominating set in* G'*. So* G' *has a dominating set of size* $k+1$*.*

($\Leftarrow$*) Suppose* G' *has a dominating set* S *of size* $k+1$*. Take* C *to be those vertices in* S *that also correspond to vertices in* G *(the ones added in step 1).*

Then C *has size at most* k *because at least one of* x *or* y *is in* S *but is not in* C*.*

To finish we verify that C *is a cover in* G*. For any edge* (u, v)*, at least one of the vertices* uv_i *is not in* S *(because there are* $k+2$ *of them and* S *has size at most* $k+1$*). So, one of the two neighbors* $\{u, v\}$ *of* uv_i *is in* S*. Thus, one of* u *or* v *is in* C*.*

Thus, C *is a vertex cover in* G *of size* k *or less.*

Happy thanksgiving!