1. Graded Problems

1. State True/False. The set of all vertices in a graph is a vertex cover.

True. Every edge has both its incident vertices (thus at least one) in the set of all vertices.
Remark: A remark about polynomial time reductions: Recall from class that we say that $A \leq_P B$ if given a black box that solves B, we can solve A in polynomial time. A special case of polynomial time reductions (called as a Karp reduction) is when the black box to solve B is used just once as shown below (when A and B are both decision problems).

```
Solve-A(Instance-A){
   Instance-B = f(Instance-A);
  return (Solve-B(Instance-B));
}
```

Here Solve-B is the black box that solves B and f is a polynomial time algorithm. To describe a Karp reduction, we need to describe what f does, that is to which instance of B a given instance of A gets mapped to. To prove that the reduction is correct, we need to show that Instance-A is an \yes" (or 1) instance of A if and only if instance-B is an \yes" (or 1) instance of B. In this homework all our reductions are Karp reductions.

2. State True/False. If $A \leq_p B$ and A ∈NP-complete, then B ∈ NP-complete.

False.
If $A \leq_p B$ and A ∈ NP-complete, then B is not necessarily in NP-complete (since B need not be in NP).

3. Show that the independent set problem is polynomial time reducible to the Hitting Set problem (Refer to Kleinberg and Tardos, Chapter 8, Exercise5 for the definition of the Hitting Set problem).

We claim that Vertex-Cover $\leq_p$ Hitting-Set.

Given an instance (G = (V, E), k') of the Vertex-Cover problem, we map it to an instance of the Hitting-Set problem as described below. Let $E = \{e_1, e_2, \ldots, e_{|E|}\}$.Set A =V, $B_1 = e_1$, $B_2 = e_2$, B_m = em and k = k'.
Remark 1: We have implicitly set n = |V| and m = |E|.
Remark 2: An edge can be considered as a set of two vertices, thus B_i are subsets of A. The mapping is clearly polynomial time computable.

All that remains to prove is that the mapping preserves membership. That is, G has a vertex cover of size k' if and only if the hitting set instance it gets mapped to has a hitting set of size k. (Note k=k')

This follows from the fact that C ⊆ V is a vertex cover of G is and only if C is a hitting set of the corresponding instance. (A =V, $B_1 = e_1$, $B_2 = e_2$, B_m = em and k = k') (Check for your self that this is the case!).

From class, we know that Independent-Set $\leq_p$ Vertex-Cover. From the transitivity of polynomial time reductions, it thus follows that Independent-Set $\leq_p$ Hitting-Set.

4. A company makes three models of desks, an executive model, an office model and a student model. Building each desk takes time in the cabinet shop, the finishing shop and the crating shop as shown in the table below:

Type of desk	Cabinet shop	Finishing shop	Crating shop	Profit
Executive	2	1	1	150
Office	1	2	1	125
Student	1	1	.5	50
Available hours	16	16	10	

How many of each type should they make to maximize profit? Use linear programming to formulate your solution. Assume that real numbers are acceptable in your solution.

Solution:

Start by defining your variables:

x = number of executive desks made

y = number of office desks made

z = number of student desks made

Maximize $P=150x+125y+50z$.

Subject to:

$2x + y + z \leq 16$ cabinet hours

$x + 2y + z \leq 16$ finishing hours

$x + y + .5z \leq 10$ crating hours

$x\geq 0,\ y\geq 0,\ z\geq 0$

2. Practice Problems

1. Given a graph $G=(V, E)$ and a positive integer $k < |V|$. The longest-simple-cycle problem is the problem of determining whether a simple cycle (no repeated vertices) of length k exists in a graph. Show that this problem is NP-complete

Solution:

Clearly this problem is in NP. The certificate will be a cycle of the graph, and the certifier will check whether the certificate is really a cycle of length k of the given graph.

We will reduce HAM-CYCLE to this problem. Given an instance of HAM-CYCLE with graph G=(V,E), construct a new graph G'=(V', E) by adding one isolated vertex u to G. Now ask the longest-simple-cycle problem with k = |V| <|V'| for graph G'.

If there is a HAM-CYCLE in G, it will be a cycle of length k = |V| in G'.

If there is no HAM-CYCLE in G', there must not be no cycle of length |V| in G'. If there is, we know the cycle does not contain u, because it is isolated. So the cycle will contain all vertex in |V|, which is a HAM-CYCLE of the G.

The reduction is in polynomial time, so longest-simple-path is NP-Complete.

2. In the Bipartite Directed Hamiltonian Cycle Problem, we are given a bipartite directed graph G = (V; E) and asked whether there is a simple cycle which visits every node exactly once. Note that this problem might potentially be easier than Directed Hamiltonian Cycle because it assumes a bipartite graph. Prove that Bipartite Directed Hamiltonian Cycle is in fact still NP-Complete.

Solution:

i) This problem is NP. Given a candidate path, we just check the nodes along the given path one by one to see if every node in the network is visited exactly once and if each consecutive node pair along the path are joined by an edge.

ii) To prove the problem is NP-complete, we reduce Directed Hamiltonian Cycle (DHC) problem to Bipartite Directed Hamiltonian Cycle (BDHC) problem.

Given an arbitrary directed graph G, we split each vertex v in G into two vertex v_in and v_out. Here v_in connects all the incoming edges to v in G; v_out connects all the outgoing edges from v in G. Moreover, we connect one directed edge from v_in to v_out. After doing these operations for each node in G, we form a new graph G'.

Here G' is bipartite graph, because we can color each v_in "blue" and each v_out "red" without any coloring conflict.

If there is DHC in G, then there is a BDHC in G'. We can replace each node v on the DHC in G into consecutive nodes v_in and v_out, and (v_in, v_out) is an edge in G'. Then the new path is a BDHC in G'.

On the other hand, if there is BDHC in G', then there is a DHC in G. Note that if v_in is on the BDHC, v_out must be the successive node on the BDHC. Then we can merge each node pair (v_in, v_out) on the BDHC in G' into node v and form a DHC in G.

In sum, G has DHC if and only if G' has a BDHC. This completes the reduction, and we confirm that the given problem is NP-complete

3. Assume that you are given a polynomial time algorithm that decides if a directed graph contains a Hamiltonian cycle. Describe a polynomial time algorithm that given a directed graph that contains a Hamiltonian cycle, lists a sequence of vertices (in order) that form a Hamiltonian cycle.

Solution:

Let $G = (V, E)$ be the input graph. Let A be an algorithm that decides if a given directed graph has a Hamiltonian cycle. Hence $A(G) = 1$.

Pick an edge $e \in E$ and remove it from G to get a new graph $\bar{G}$.
If $A(\bar{G}) = 1$, then there exists a Hamiltonian cycle in $\bar{G}$ which is a subgraph of G, set $G = \bar{G}$.
If $A(\bar{G}) = 0$, then every Hamiltonian cycle in G contains e. Put e back into G.

Iterate the above three lines until we are left with exactly $|V|$ edges. Since after each step we are left with a subgraph that contains a Hamiltonian cycle, at termination we are left with the set of edges that forms a Hamiltonian cycle. Starting from an edge, do a BFS to enumerate the edges of the Hamiltonian cycle in order.