The "Every Problem Gets a Hint" Problem Set

Question 1. *Given a directed graph $G = (V, E)$ with a distinguished vertex $s \in V$ (s for "start"), consider the following 2-person game on G in which players alternate moves: Player 1 begins by placing a stone on vertex s, after which Player 2 can move that stone to any "out" neighbor of s (that is, to any u such that $(s, u) \in E$). After these initial moves, players continue to alternate turns, at each turn moving the stone from its current location x to a new location y such that $(x, y) \in E$. There is one additional rule: the stone can never return to a vertex it has previously visited. A player loses if, when it becomes their turn, they have no move (all out neighbors of the vertex upon which the stone currently rests have already been visited). For some instances (G, s) of this game, Player 1 may be able to force a win (that is, always be able to win regardless of the moves that Player 2 makes), and for some instances, Player 2 might be able to force a win. It turns out that determining whether a player can force a win is NP-hard. This problem appears to be so hard, in fact, that it is not generally believed to be in the class NP!*

Your task: Show that if G is a DAG (directed acyclic graph) then one can determine, in polynomial time, whether one of the players has a winning strategy.

Hint: Sinks are inherently losing positions. Perhaps you can label each node as inherently winning or losing...?

Question 2. *Chapter 10, Problem 1. Hint: Try the idea we used for vertex cover (see Property 10.3): Delete some element of some B_i (and B_i) from the problem instance Think about how this reduction to a smaller problem can help.*

Question 3. *Chapter 10, Problem 2. (a) Hint: See if you can figure out how to break the problem into subproblems that give the recursion $T(n, d) = 3T(n, d-1) + p(n)$ for some polynomial $p()$; then solve the recurrence. (b) Hint: Think about the two assignments that set every variable to 0 and every variable to 1, respectively. How far can an arbitrary assignment Φ simultaneously be from these two assignments? Divide and conquer....*

Question 4. *Chapter 10, Problem 3. Hint: There is a Hamiltonian path from v_1 to v_n if and only if, for some v_i there is an edge from v_i to v_n (final edge) and a Hamiltonian path (in $G - \{v_n\}$) from v_1 to v_i. This suggests dynamic programming: if we had a table $H[S, 1, j]$ to tell us if there is a Hamiltonian path from v_1 to v_j using only the vertices of S, we'd be set! [Google "xkcd #399" for more information.]*