A. Perform the reduction from 3SAT to HAMPATH (Theorem 7.46) on these examples:

$$(x \vee y \vee \bar{z}) \wedge (\bar{x} \vee \bar{y} \vee \bar{z}), \qquad (x \vee y \vee y) \wedge (x \vee \bar{y} \vee \bar{y}) \wedge (\bar{x} \vee \bar{x} \vee \bar{x}).$$

Based on these examples, explain how if ϕ has a satisfying assignment then the resulting graph G has a Hamiltonian path from node s to node t. Then explain the converse.

B. Suppose that a language A is decidable in space $s(n)$ on a non-deterministic Turing machine. Estimate the time required to decide A on a deterministic Turing machine. (I can think of two ways to proceed. Maybe there are others. The approach that produces the lowest bound earns the most credit.)

C. Let $A = \{\langle M \rangle : M \text{ is a TM with time complexity } t(n) = n^2\}$. Show that A is not recognizable. (Hint: $\text{HALT}_{\text{TM}}{}^c$.)

D. The following context-free grammar is in Chomsky normal form. Execute the dynamic programming polynomial-time algorithm for deciding whether the input $w = aabba$ is derivable from the grammar. (This task takes me less than two minutes.)

$$S \to RT \qquad R \to TR \mid a \qquad T \to TR \mid b$$

E. Is MAX-CUT NP-complete?

The rest of this review is about material from earlier in the course.

On Day 14, you proved that the class of decidable languages is closed under various operations, including complementation. You also proved that the class of recognizable languages is closed under various operations, not including complementation.

F.A. Why doesn't your proof for the decidable case also work for the recognizable case? How does that proof break?

F.B. Anyway, is the class of recognizable languages closed under complementation?

G. Let A be the set of all strings $\langle M, w \rangle$, where M is a Turing machine, w is an input for M, and M tries to move its tape head off the left end of its tape sometime during its computation on input w. Show that A is undecidable.

H. Let A be a finite language. Then A is regular, so A has a pumping length p (for use in the pumping lemma for regular languages). What is that p?