This homework is due Thursday March 1 at the start of class. Questions are drawn from the material in Sections 7.4 and 7.5 of the text on NP-completeness.

To show that a language is decidable in deterministic polynomial time, you may present a decision *algorithm* that runs in polynomial time in the standard unit-cost random-access machine model.

The homework is worth 100 points. When questions with several parts do not specify the points for each part, each part has equal weight.

Remember to write on just one side of a page, do not use scrap paper, put your answers in the correct order, and staple your pages together. If you can't solve a problem, state this, and write only what you know to be correct. Neatness and conciseness count.

(1) **(Long paths)** (10 points) A *simple* path in an undirected graph is a path that does not contain a cycle. A path from vertex s to vertex t is an (s,t)-*path*. In a graph whose edges have associated integer lengths, the *length* of a path is the sum of the lengths of its edges. Let

$$\text{LongPath} := \Big\{\langle G,s,t,k\rangle \,:\, G \text{ has a simple } (s,t)\text{-path of length at least } k\Big\}.$$

Prove that LongPath is NP-complete.

(Hint: You may assume that HamiltonianPath on undirected graphs is NP-complete.)

(Note: Whether an undirected graph has a simple (s,t)-path of length *at most* k can be decided in polynomial time.)

(2) **(Two satisfying assignments)** (5 points) Let

$$\text{DoubleSat} := \left\{\langle F\rangle \,:\, \begin{pmatrix} F \text{ is a Boolean formula that has} \\ \text{at least two satisfying assignments}\end{pmatrix}\right\}.$$

Prove that DoubleSat is NP-complete.

(3) **(Three literals per variable)** (20 points) Let

$$\text{TripleSat} := \left\{\langle F\rangle \,:\, \begin{pmatrix} F \text{ is a satisfiable Boolean formula in conjunctive normal} \\ \text{form where each variable occurs in at most three literals}\end{pmatrix}\right\}.$$

Prove that TripleSat is NP-complete.

(Hint: TripleSat is not restricted to have three literals per clause, which can simplify a reduction from ThreeSat.)

(Note: Whether a formula whose variables occur in at most *two* literals is satisfiable can be decided in polynomial time.)

(4) **(Finding a satisfying assignment)** (20 points) Satisfiability is the problem of *deciding* whether a Boolean formula has a satisfying assignment. Suppose Satisfiability can be solved in polynomial time. Show that in this case, *finding* a satisfying assignment can also be solved in polynomial time.

(Hint: Give an algorithm that by making repeated calls to a decision algorithm for Satisfiability on modified formulas, outputs a satisfying assignment if one exists.)

(5) **(Dominating set)** (15 points) In an undirected graph, a vertex subset D is a *dominating set* if every vertex outside D has an edge to some vertex in D. Let

$$\text{DominatingSet} := \left\{ \langle G, k \rangle : \begin{pmatrix} G \text{ is an undirected graph that contains} \\ \text{a dominating set of at most } k \text{ vertices} \end{pmatrix} \right\}.$$

Prove that DominatingSet is NP-complete.

(Hint: You may assume that VertexCover is NP-complete.)

(6) **(Scheduling)** (15 points) Suppose there is a set of final exams $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$ and a set of students $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, where each student is represented by the subset of the final exams $S_i \subseteq \mathcal{E}$ that the student needs to take. Given a number $k \leq m$ of distinct exam times, the exams can be scheduled *without conflicts* if a time can be assigned to each exam so that no student has to take two or more exams at the same time. Let

$$\text{Scheduling} := \left\{ \langle \mathcal{E}, \mathcal{S}, k \rangle : \begin{pmatrix} \mathcal{E} \text{ can be scheduled at } k \text{ times so that} \\ \text{no student in } \mathcal{S} \text{ has an exam conflict} \end{pmatrix} \right\}.$$

Prove that Scheduling is NP-complete.

(Hint: You may assume that ThreeColor, defined in Problem 7.27 of the text, is NP-complete.)

(7) **(Set splitting)** (15 points) Suppose there is a ground set $\mathcal{S} = \{x_1, x_2, \ldots, x_n\}$ and a family of subsets $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ where each $F_i \subseteq \mathcal{S}$. A *k-coloring* of $\mathcal{S}$ is a labeling of the elements of $\mathcal{S}$ using at most k colors. A coloring of $\mathcal{S}$ is said to *split* the family $\mathcal{F}$ of sets if no subset $F_i \in \mathcal{F}$ has all its elements labeled by the same color. Let

$$\text{SetSplitting} := \left\{ \langle \mathcal{S}, \mathcal{F} \rangle : \mathcal{S} \text{ has a two-coloring that splits } \mathcal{F} \right\}.$$

Prove that SetSplitting is NP-complete.

(Hint: You may assume that UnequalSat, defined in Problem 7.24 of the text, is NP-complete.)