18.404/6.5400 Michael Sipser
Theory of Computation
Due via Gradescope at noon sharp, Thursday, November 9, 2023 **Problem Set 4**

Read all of Chapter 7.

1. Let $MODEXP = \{\langle a, b, c, p\rangle \mid a, b, c, p$ are positive binary integers such that $a^b \equiv c \pmod{p}\}$. Show that $MODEXP \in \mathrm{P}$. (Hint: Try $b = 8$. You can assume that basic arithmetical operations, such as $+$, $\times$, and mod, are computable in polynomial time.)

2. Let $\#PATH = \{\langle G, s, t, k\rangle \mid$ directed graph G has exactly k paths from s to $t\}$. Here, paths aren't required to be simple, i.e., they may repeat nodes. Show that $\#PATH \in \mathrm{P}$.

3. Show that if P = NP, then every language $A \in \mathrm{P}$, except $A = \emptyset$ and $A = \Sigma^*$, is NP-complete.

4. Show that if P = NP, a polynomial time algorithm exists that produces a satisfying assignment when given a satisfiable Boolean formula. (Note: The algorithm you are asked to provide computes a function, and NP contains languages, not functions. The P = NP assumption implies that SAT is in P, so testing satisfiability is solvable in polynomial time. But the assumption doesn't say how this test is done, and the test may not reveal satisfying assignments. You must show that you can find them anyway. Hint: Use the satisfiability tester repeatedly to find the assignment bit-by-bit.)

5. In a directed graph, the ***indegree*** of a node is the number of incoming edges and the ***outdegree*** is the number of outgoing edges. Show that the following problem is NP-complete. Given an undirected graph G and a designated subset C of G's nodes, is it possible to convert G to a directed graph by assigning directions to each of its edges so that every node in C has indegree 0 or outdegree 0, and every other node in G has indegree at least 1.

6. This problem is inspired by the single-player game *Minesweeper*, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden *mine* or is empty. The player chooses nodes, one by one. If the player chooses a node containing a mine, the player loses. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (A neighboring node is one connected to the chosen node by an edge.) The player wins if and when all empty nodes have been so chosen.

   In the *mine consistency problem*, you are given a graph G along with numbers labeling some of G's nodes. You must determine whether a placement of mines on the remaining nodes is possible, so that any node v that is labeled m has exactly m neighboring nodes containing mines. Formulate this problem as a language and show that it is NP-complete.

7.* (optional) The ***difference hierarchy*** $\mathrm{D}_i\mathrm{P}$ is defined recursively as

   i. $\mathrm{D}_1\mathrm{P} = \mathrm{NP}$, and
   ii. $\mathrm{D}_i\mathrm{P} = \{A \mid A = B \setminus C$ for B in NP and C in $\mathrm{D}_{i-1}\mathrm{P}\}$. (Here $B \setminus C = B \cap \overline{C}$.)

   For example, a language in $\mathrm{D}_2\mathrm{P}$ is the difference of two NP languages. Let $\mathrm{DP} = \mathrm{D}_2\mathrm{P}$. Let

$$Z = \{\langle G_1, k_1, G_2, k_2\rangle \mid G_1 \text{ has a } k_1\text{-clique and } G_2 \text{ doesn't have a } k_2\text{-clique}\}.$$

a. Show that Z is complete for DP. In other words, show that Z is in DP and every language in DP is polynomial time reducible to Z.

b. Let $MAX\text{-}CLIQUE = \{\langle G, k\rangle \mid$ a largest clique in G is of size exactly $k\}$.
Use part (a) to show that $MAX\text{-}CLIQUE$ is DP-complete.