We may grade a **subset of the assigned questions**, to be determined after the deadline, so that we can provide better feedback on the graded questions.
Unless otherwise stated, each question requires sufficient justification to convince the reader of the correctness of your answer.
For bonus questions, we will not provide any insight during office hours or Piazza, and we do not guarantee anything about the difficulty of these questions.
We strongly encourage you to typeset your solutions in LaTeX.
If you collaborated with someone, you must state their name(s). You must write your own solution for all problems and may not look at any other student's write-up.

0. If applicable, state the name(s) and uniqname(s) of your collaborator(s).

Solution:

1. (a) Prove the transitive property for polynomial-time mapping reductions:

$$A \leq_p B \text{ and } B \leq_p C \implies A \leq_p C.$$

   (b) Prove that if $A \leq_p B$ and $B \in \mathsf{NP}$, then $A \in \mathsf{NP}$.
   (c) Prove that if $A \leq_p B$, then $A \leq_T B$.

Solution:

2. Recall the 0-1 Knapsack Problem: we are given two length-n arrays containing positive integer weights $W = (w_1, w_2, \ldots, w_n)$ and values $V = (v_1, v_2, \ldots, v_n)$, and a weight capacity $C \in \mathbb{N}$. The goal is to select items having maximum total value whose total weight is below the threshold. We can define this as a decision problem by introducing a threshold K and asking whether a value of at least K can be achieved (subject to the weight constraint):

$$\textsc{Knapsack} = \{(W, V, C, K) : \exists\, S \subseteq \{1, \ldots, n\} \text{ such that } \sum_{i \in S} W[i] \leq C \text{ and } \sum_{i \in S} V[i] \geq K\}.$$

   In this problem you will show that Knapsack is NP-Complete.

   (a) Prove that $\textsc{Knapsack} \in \mathsf{NP}$.
   (b) Prove that Knapsack is NP-Hard, by showing that $\textsc{Subset-Sum} \leq_p \textsc{Knapsack}$. Conclude that Knapsack is NP-Complete.
   (c) Recall that we previously gave a dynamic programming algorithm that solves Knapsack in $O(nC)$ time. Does this prove that $\mathsf{P} = \mathsf{NP}$? Why or why not?

Solution:

3. Recall that CLIQUE $\in$ NP, where CLIQUE is defined as:

$$\text{CLIQUE} = \{\langle G, k\rangle : G = (V, E) \text{ has a clique of size} \geq k\}.$$

(Recall: A *clique* of an undirected graph $G = (V, E)$ is a subset $K \subseteq V$ such that every pair of vertices in K has an edge between them.) Consider the following variant of CLIQUE:

$$\text{HALF-CLIQUE} = \{\langle G\rangle : |V| = 2n, \text{ and } G \text{ has a clique of size} \geq n\}.$$

Prove that HALF-CLIQUE is NP-complete.

Solution:

4. Suppose there are n students at Michigan and k clubs. Every student may join any number of clubs, possibly zero. Let $S = (S_1, \ldots, S_k)$ be a list of the students in each club, where each club i has a subset $S_i \subset [n]$ of students.

Now given a tuple $q = (q_1, \ldots, q_k)$ of natural numbers, we want to enroll a subset of Michigan students in EECS 376 so that there are exactly q_i EECS 376 students in the ith club, for every $i \in [k]$.

We say q is a *possible configuration* if this is possible. Note that not all qs are possible configurations. For example, if $S_i = [n]$ for every $i \in [k]$, i.e. every student joins every club, then the only possible q's are $(w, w, \ldots, w)$ for some $w \in \{0, 1, \ldots, n\}$ where w is the number of EECS 376 students.

Concretely, define

$$\begin{aligned}\text{POSS-CONFIG} = \{\langle n, k, S, q\rangle : {} & q \text{ is a possible configuration, i.e. there exists } E \subset [n], \\ & \text{representing the set of EECS 376 students, such that} \\ & \text{for every } i \in [k],\ |S_i \cap E| = q_i\}.\end{aligned}$$

Prove that 3SAT $\leq_p$ POSS-CONFIG.

Hints:

(a) For each variable v_i in φ, allocate two students to represent when $v_i = T$ and $v_i = F$, respectively. How can you enforce that exactly one of them is enrolled in EECS 376 (perhaps by definition of a corresponding club)?

(b) Assuming you've dealt with hint (a), how do you capture the constraint that all clauses evaluate to true (again by definition of a corresponding club)?

Solution:

5. Let EXP be the class of all languages which are decidable in exponential time, i.e., in time $O(2^{n^k})$ for some constant k (where n is the length of input). Formally,

$$\mathsf{EXP} = \bigcup_{k \in \mathbb{N}} \mathsf{DTIME}\left(2^{n^k}\right).$$

It remains unknown whether $\mathsf{NP} = \mathsf{EXP}$, but it is known that $\mathsf{P} \neq \mathsf{EXP}$. Prove that $\mathsf{NP} \subseteq \mathsf{EXP}$.

That is, for any language $L \in \mathsf{NP}$ and any input $x \in L$, provide a decider that runs in time $O(2^{n^k})$, where $n = |x|$ and k is some constant. Recall that any language $L \in \mathsf{NP}$ has a verifier $V(x, c)$ which runs in time $O(|x|^d)$ for a constant d. (As always, you should analyze the correctness and runtime of your decider.)

Solution: