CS 170 Homework 13

Due **Monday 4/20/2026, at 10:00 pm (grace period until 11:59pm)**

Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, explicitly write "none".

1 NP Basics (Solo Question; 10 points)

Assume A reduces to B in polynomial time. In each part you will be given a fact about one of the problems. What information can you derive of the other problem given each fact?

(a) A is in **P**.

(b) B is in **P**.

(c) A is **NP**-hard.

(d) B is **NP**-hard.

2 TSP Variants (10 points)

In the Traveling Salesman Problem (TSP), we are given an undirected graph G with non-negative weights and asked to find a minimum weight cycle that visits each vertex exactly once.

In the s-t Traveling Salesman Problem (s-t TSP), we are given an undirected graph G with non-negative weights, two vertices s and t, and are asked to find a minimum weight path that starts at s, visits all other vertices exactly once, and ends at t.

(a) Consider the following reduction from s-t TSP to TSP: Take G and add a weight 0 edge between s and t to get a new graph G'. Show that if a s-t TSP solution of weight w exists in G, then a TSP solution of weight w exists in G'.

(b) Despite what you proved in part (a), why is the reduction in part (a) not valid?

(c) Give a valid reduction from s-t TSP to TSP. Prove correctness for your reduction. No runtime analysis needed.

3 SAT and Integer Programming (10 points)

Recall the following problems:

1. **3-SAT**: Given a set of boolean variables x_i with a set of clauses, where each clause is the OR of at most 3 literals, we want to decide if there exists an assignment of the variables that satisfies all the clauses.

   Recall that a literal is a variable or its negation; an example clause is $x_1 \vee \overline{x_4} \vee \overline{x_7}$.

2. **Integer linear programming feasibility**: Given a set of variables x_i with a set of constraints, we want to decide if there exists an assignment of the variables that satisfies all the constraints. Each constraint is either a linear inequality, or a 0-1 constraint $x_i \in \{0, 1\}$. (We are not given an objective function.)

Give a reduction from 3-SAT to integer linear programming feasibility, and briefly justify its correctness.

4 Algorithms for NP Problems (10 points)

(a) Give a $O(2^n m)$-time algorithm to solve a 3-SAT instance with n variables and m constraints (see Question 3).

(b) Show that every problem in **NP** can be solved in time $O(2^{n^c})$ for some constant c (that can depend on the problem). Here n denotes the size of (i.e. the number of bits needed to write down) the original problem instance.

*Hint: use part (a) along with the fact that 3-SAT is **NP**-hard.*

5 Halting (10 points)

The *halting problem* has input given by a program (as e.g. a .py file), and asks us to determine if the program runs forever or eventually halts.

(a) Give a reduction from 3-SAT (or another **NP**-complete problem) to the halting problem. Your reduction algorithm, which *constructs* a program from the 3-SAT instance, should run in polynomial time, but the resulting program may run for arbitrarily long.

(Your reduction will imply that the halting problem is **NP**-hard.)

(b) Read the box on page 262 of the textbook DPV, which shows that there is no algorithm that solves the halting problem in finite time. Using this fact, can you conclude whether or not the halting problem is in **NP**?

Hint: it may be helpful to recall your solution to Question 4.

