CSCI 432/532, Spring 2025

Homework 6

Due Monday, March 3, 2025 at 11:59pm Mountain Time

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF to the appropriate Canvas dropbox.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document.
- When possible, the homework will include at least one fully solved problem, similar to that week's assigned problems, together with the rubric we would use to grade this problem if it appeared in an actual homework or exam. These model solutions show our recommendations for structure, presentation, and level of detail in your homework solutions. (Obviously, the actual *content* of your solutions won't match the model solutions, because your problems are different!)

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you must

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on problem 2. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve problem 1 part (c); "I put my solution for problem 1 part (c) into ChatGPT to check that it was correct and it caught a missing case and suggested some grammer fixes.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading Rubrics

For problem 1, either prove by diagonalization or by reduction, with different rubrics for each.

Undecidability by diagonalization.
+ 4 for correct wrapper Turing machine
+ 6 for self-contradiction proof (= 3 for $\Leftarrow$ + 3 for $\Rightarrow$)

Undecidability by reduction.
+ 4 for correct reduction
+ 3 for “if” proof
+ 3 for “only if” proof

For problems 2 and 3:

Basic reduction (algorithms). 10 points =

+ 4 for calling the magic black box as a subroutine in the algorithm. No points here if it is not called on the correct input data type.
+ 2 for a correct algorithm.
+ 2 for an algorithm that runs in polynomial time (assuming the black box runs in polynomial time).
+ 2 for a valid argument that the algorithm runs in polynomial time.

1. Let $\langle M \rangle$ denote the string encoding of a Turing machine M. Recall that if w is a string, then ww is a new string that repeats w twice. Prove that the following language is undecidable. You may use either a diagonalization or reduction proof.

$$\textsc{SelfRepeatAccept} := \{\langle M \rangle \mid M \text{ accepts the string } \langle M \rangle \langle M \rangle\}$$

2. Recall that a ***vertex cover*** of a graph is a set of a vertices that touches every edge of that graph. Suppose you are given a magic black box that somehow answers the following decision problem *in polynomial time*:

   - Input: an undirected graph G and a positive integer k.
   - Output: True if G has a vertex cover of size k and False otherwise.

   (a) Using this black box as a subroutine, describe an algorithm that solves the following optimization problem *in polynomial time*:
      - Input: an undirected graph G.
      - Output: the size of the smallest vertex cover of G.

   (b) Using this black box as a subroutine, describe an algorithm that solves the following optimization problem *in polynomial time*:
      - Input: an undirected graph G.
      - Output: a vertex cover in G of smallest size.

   Unlike in the problem session solutions, you do not need to argue the correctness of your algorithms. But you still do need to argue that they run in polynomial time.

3. Formally, a ***proper coloring*** of a graph $G = (V, E)$ is a function $c : V \to \{1, 2, \ldots, k\}$, for some integer k, such that $c(u) \neq c(v)$ for all $uv \in E$. Less formally, a valid coloring assigns each vertex of G a color, such that every edge in G has endpoints with different colors. The ***chromatic number*** of a graph is the minimum number of colors in a proper coloring of G.

   Suppose you are given a magic black box that somehow answers the following decision problem *in polynomial time*:

   - Input: An undirected graph G and an integer k.
   - Output: True if G has a proper coloring with k colors, and False otherwise.

   Using this black box as a subroutine, describe an algorithm that solves the following ***coloring problem*** *in polynomial time*:

   - Input: An undirected graph G.
   - Output: A valid coloring of G using the minimum possible number of colors.

   *[Hint: You can use the magic box more than once. The input to the magic box is a graph and **only** a graph, meaning **only** vertices and edges.]*

   Unlike in the problem session solutions, you do not need to argue the correctness of your algorithms. But you still do need to argue that they run in polynomial time.