CSCI 432/532, Spring 2024

Homework 7

Due Monday, March 4, 2024 at 9pm 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 on Gradescope.
- 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.
- Each 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. 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!) Note: this copy currently does not have any solved problems, but will updated with solved problems soon.

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 1(e). Note that many of these come for free when you fix the broken proof that is already provided in (c)!

NP-hardness proof. 10 points =

+ 1 point for choosing a reasonable NP-hard problem X to reduce from.
    - The Cook-Levin theorem implies that *in principle* one can prove NP-hardness by reduction from *any* NP-hard problem. What we're looking for here is a problem where a simple and

direct NP-hardness proof seems likely.

- You can use any of the NP-hard problems listed in the lecture notes (except the one you are trying to prove NP-hard, of course).

+ 2 for a *structurally sound* polynomial-time reduction. Specifically, the reduction must:

  - take an *arbitrary* instance of the declared problem X ***and nothing else*** as input,
  - transform that input into a corresponding instance of Y (the problem we're trying to prove NP-hard),
  - transform the output of the oracle for Y into a reasonable output for X, and
  - run in polynomial time.

  (The output transformation is usually trivial.) This is strictly abou tthe structure of the reduction algorithm, not about its correctness. No credit for the rest of the problem if thi is wrong.

+ 2 points for a *correct* polynomial-time reduction. That is, assuming a black-box algorithm that solves Y in polynomial time, the proposed reduction actually solves problem X in polynomial time.

+ 2 points for the "if" proof of correctness. (Every good instance of X is transformed into a good instance of Y).

+ 2 points for the "only if" proof of correctness. (Every bad instance of X is transformed into a bad instance of Y—note that you may prove this by proving that if your transformation produces a good instance of X then it was given a good instance of Y).

+ 1 for writing "polyomial time".

- An incorrect but structurally sound polynomial-time reduction that still satisfies half of the correctness proof is worth at most 5/10 (=1 for reasonable reduction source + 2 for structural soundness +2 for the half of the proof).
- A reduction in the wrong direction is worth at most 1/10 (for choosing a reasonable problem).

1. As we saw in the problem session, the 3Color problem takes in a graph and asks whether there is a way to assign each vertex one of three colors so that every edge has endpoints of a different color. We call such a coloring a *full 3-coloring*. Similarly, a 3-coloring of a graph G is called an *almost 3-coloring* if every vertex has at most one neighbor with the same color. The Almost3Color problem asks whether, given graph G, it has an almost 3-coloring.

   (a) Is Almost3Color in NP? How do you know? (2 points)

   (b) Draw an example graph that has an almost 3-coloring but not a full 3-coloring. (2 points)

   (c) Here is a proposed proof that Almost3Color is NP-hard.

   > **Solution:** We reduce from 3Color. Given an arbitrary input graph G, we construct a new graph H by attaching a clique of 4 vertices to every vertex of G. Specifically, for each vertex v in G, the graph H contains three new vertices v_1, v_2, v_3, along with edges $vv_1, vv_2, vv_3, v_1v_2, v_1v_3, v_2v_3$. Notice that this transformation creates $4|V|$ new vertices and $6|V|$ new edges, so it can be done in polynomial time.
   >
   > Now, I claim that G has a full 3-coloring if and only if H has an almost 4-coloring. I show both directions of the implication.
   >
   > $\Rightarrow$ Suppose G has a full 3-coloring using the colors red, yellow, and blue. Extend this color assignment to the vertices of H by coloring each vertex v_1 red, each vertex v_2 yellow, and each vertex v_3 blue. With this assignment, each vertex of H has at most one neighbor of the same color. Specifically, each vertex of G has the same color as one of the vertices in this gadget, and the other two vertices in v's gadget have no neighbors with the same color.
   >
   > $\Leftarrow$ Now suppose H has an almost 3-coloring. Then G must have a full 3-coloring because...um... ■

   Give a graph G such that G does not have a full 3-coloring but the graph H constructed by this reduction does have an almost 3-coloring. (5 points)

   (d) Describe a small graph X with the following property: In every almost 3-coloring of X, every vertex of X has *exactly* one neighbor with the same color. (2 points)

   (e) Using your graph X to change the transformation above, give a full, correct proof of that SlightlyImproper3Color is indeed NP-hard. (10 points—see rubric)