

CS 3510: Homework 5

Due on Thursday, July 15, 11:59pm via Gradescope.
Late submission with no penalty until Friday, July 16, 11:59am.

Professor Brito

CS 3510 Staff

Instructions.

For the graded problems, you are allow to declare the following problems in the class NP-hard without further justification.

- SAT, 3-SAT, IS, Clique, Vertex Cover, Rudrata Path.

The steps to write a reduction.

In a reduction, we have a Problem B, and we want to show that it is NP-Complete. These are the steps:

- Demonstrate that problem B is in the class of NP problems. This is the design of the verifier. More precisely, given an instance and a candidate solution, describe an algorithm that verifies if the candidate solution is indeed a valid solution. It needs to address the runtime.
- Demonstrate that problem B is at least as hard as a problem known to be NP-Complete or NP-Hard. This is done via reduction from the known problem A to the unknown problem B ($A \rightarrow B$) as follows:
 1. Show how an instance of A is converted to an instance of B in polynomial time.
 2. Show how a solution to the instance of B can be converted to a solution for the initial instance of A, again in polynomial time.
 3. Furthermore, you must show that a solution for B exists if-and-only-if (IFF) a solution to A exists. This means that, after proving the previous step, you are left with checking that if there is no solution for B, then no solution exists for A. This may be done via the contra-positive approach, showing that if a solution exists for A then a solution for B must exist.

The second bullet point above is the proof that problem B is NP-hard which combined with the first bullet point yields the NP-complete proof.

Suggested reading.

We are covering topics from Chapter 8. You do not need to cover reductions we did not see in lecture.

Suggested problems.

All problems are from Chapter 8. The numeration corresponds to the 2006 edition.

- 8.1 (this problem can help you think how many optimization problems can be modified to search problems and how to reduce from one to another)
- Show that Clique is NP-hard by reducing Clique-search to it.
- 8.3 (Stingy-SAT)
- 8.6, 8.8, 8.9 (Hitting-Set), 8.10 (proof by generalization), 8.12, 8.13 (multiple tree problems), 8.14, 8.15, 8.19 (Kite), 8.20 (Dominating-Set), 8.22 (Feedback-arc-Set).

Practice problem.

The Dense Subgraph Problem takes as input an undirected graph $G = (V, E)$ and two positive integers a and b , and returns a subset of *exactly* a vertices such that there are *at least* b edges connecting them, if such set exists, or return NO otherwise.

Show that the Dense Subgraph Problem is NP-complete.

Hint: this problem is one of the parts of problem 8.10 (proof by generalization). We advice you to write a full reduction.

DO NOT USE PSEUDOCODE

A solution using pseudocode will receive a zero, even if it is correct.

Problem 1

Show that the following problem is **NP-complete**.

Input: A boolean function in CNF such that every clause has at most three literals and every variable appears in at most three clauses.

Output: An assignment that evaluates the given function TRUE, if such assignment exists, and report NO otherwise.

Hint: the solution is outlined in the book! Make sure to fill the gaps left in the book when you answer your homework.

Problem 2

The **Max-diameter-spanning-tree** problem receives as input an undirected, connected graph, and a natural number $g > 0$, and outputs a spanning tree of diameter greater or equal than g or report NO if such tree does not exist. Prove that this problem is NP-HARD. (The diameter of a graph is the length of the longest path.)