CS-E4530 Computational Complexity Theory, Spring 2020
Problem Set 5 **Homework Due Tue 25 Feb, 23:59**

Note: Because of the exam week, there are no lectures or tutorial sessions on 17–21 February. Teaching resumes again normally in the following week. The 1st midterm exam is on Mon 17 Feb, 9–12 a.m., Lecture Hall T1.

Instructions

- Classroom Problems C5.1–C5.3 will be discussed and solved at the tutorial session on Wed 12 Feb, 14–16, Room T4 (A238). No credit is given for these problems.
- Homework Problems H5.1–H5.3 you should solve on your own, and submit your solutions via the MyCourses interface by the deadline of Tue 25 Feb, 23:59, These problems will be individually graded on a scale of 0–2 points per problem.
- In preparing your solutions to the Homework Problems:
  1. Justify your solutions, be precise, and provide sufficient detail so that it is easy to follow your reasoning.
  2. Submit your solutions as an easily readable, single pdf file, which is either typeset or written in full sentences and clean handwriting.
  3. **[Code of Conduct]** You can discuss the problems with your colleagues and the course's teaching staff, but you must write the presentations of your solutions *independently* and *individually*, without any notes from such discussions.

Classroom Problems

C5.1 Prove that the following decision problem TMSAT is **NP**-complete:

> INSTANCE: A tuple $(M, x, 1^n, 1^t)$, where M is a Turing machine code and $x \in \{0,1\}^*$. (In some appropriate tuple encoding.)
> QUESTION: Is there a string $u \in \{0,1\}^*$ with $|u| \leq n$ such that the Turing machine M accepts input (x, u) within t steps?

C5.2 Show that the following LONGEST PATH problem in **NP**-complete:

> INSTANCE: An undirected graph $G = (V, E)$ and an integer K.
> QUESTION: Does G contain a simple path (that is, a path encountering no vertex more than once) with K or more edges?

C5.3 Establish the **NP**-completeness of the INTEGER PROGRAMMING problem (Lecture 9, Theorem 9.22) by a reduction from the 3SAT problem.

CS-E4530 Computational Complexity Theory, Spring 2020
Problem Set 5 **Homework Due Tue 25 Feb, 23:59**

Homework Problems

H5.1 Show that the MAX CUT problem (Lecture 8, Theorem 9.12) is **NP**-complete also for simple graphs, i.e. graphs with no multiedges and no self-loops. (*Hint:* Reduction from NAESAT. For a NAESAT formula ϕ with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, construct a graph G_ϕ on $6mn$ vertices, $3m$ for each positive literal x_i and $3m$ for each negative literal $\neg x_i$. Follow the proof scheme of Theorem 9.12, but use different literal vertices in different clause triangles.) (2 points)

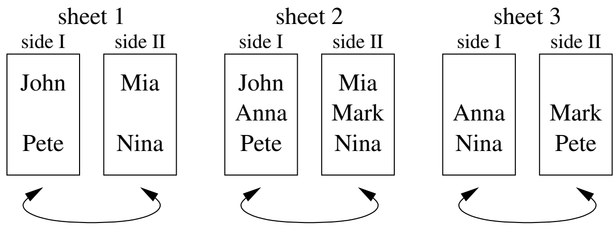
H5.2 Show that the following CLIQUE COVER problem is NP-complete:

> INSTANCE: An undirected graph $G = (V, E)$ and an integer K.
> QUESTION: Can the vertices of G be partitioned into at most K sets $V_1, \ldots, V_K$, so that each V_i forms a clique in G?

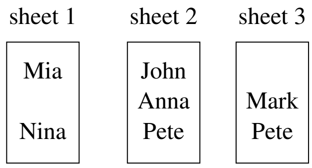
(*Hint:* Reduction from GRAPH COLOURING.) (2 points)

H5.3 Consider the following MENTION ALL problem. As an instance, you are given a set of names and a set of sheets of paper. Each sheet of paper contains some of the names written on one side and some other names (not necessarily different ones) written on the other side. Either side of a sheet can also be empty. The question is: can we place the sheets on a table so that every name shows up?

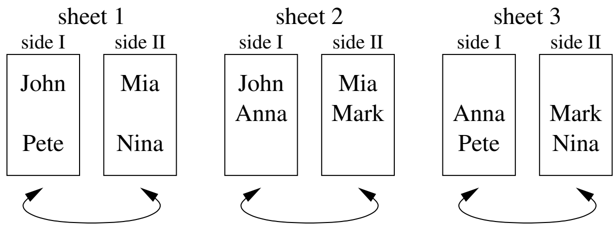
Suppose, for instance, that we have the names {Anna, John, Mark, Mia, Nina, Pete} and the following sheets:



Now we have the solution:



As another example, if we have the names {Anna, John, Mark, Mia, Nina, Pete} and the sheets



then no solution is possible.

Show that the MENTION ALL problem is **NP**-complete. (2 points)