601.433/633 Introduction to Algorithms **Fall 2024**
Homework #9 **Due:** December 5, 2024, 9am

Remember: you may work in groups of up to three people, but must write up your solution entirely on your own. Solutions must by typeset (LaTeX preferred but not required). Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. You may use the internet to look up formulas, definitions, etc., but may not simply look up the answers online.

Please include proofs with all of your answers, unless stated otherwise.

1 Graduation Requirements (33 points)

John Hopskins University[1] has n courses. In order to graduate, a student must satisfy several requirements of the form "you must take at least 1 course from subset S". Moreover, courses *are allowed* be used towards multiple requirement. For example, if there was a requirement that a student must take at least one course from $\{A, B, C\}$, and another required at least one course from $\{C, D, E\}$, and a third required at least one course from $\{A, F, G\}$, then a student would only have to take A and C to graduate.

Now consider an incoming freshman interested in finding the *minimum* number of courses required to graduate. Your job is to prove that the problem faced by this freshman is NP-complete. More formally, consider the following decision problem: given n items (say $a_1, \ldots a_n$), given m subsets of these items $S_1, S_2, \ldots, S_m$, and given an integer k, does there exist a set S of at most k items such that $|S \cap S_i| \geq 1$ for all $i \in \{1, \ldots, m\}$.

(a) (11 points) Prove that this problem is in NP.

(b) (22 points) Prove that this problem is NP-hard.

2 Magic Subroutines (34 points)

(a) (17 points) Suppose you are given a magic black box that, given an arbitrary graph G, can determine (in polynomial time) the number of vertices in the largest clique in G. Describe a polynomial-time algorithm which uses this magic black box as a subroutine and that, given an arbitrary graph G, returns a clique of G of maximum size. Prove polynomial running time and correctness.

(b) (17 points) Suppose you are given a magic black box that, given an arbitrary boolean circuit Φ (with one output and no loops, like in Circuit-SAT), can determine in polynomial time whether Φ is satisfiable. Describe a polynomial-time algorithm that either computes a satisfying input for a given boolean circuit or correctly reports that no such input exists, using the magic black box as a subroutine. Prove polynomial running time and correctness.

[1] https://www.youtube.com/watch?v=JEH2ha1p0WA

3 Integer Linear Programming (33 points)

In class we talked about linear programming, and the fact that it can be solved in polynomial time. Slightly more formally, we defined the feasibility version of linear programming to be the following decision problem

- Input: n variables $x_1, x_2, \ldots, x_n$, and m non-strict linear inequalities over the variables.
- Output: YES if there is a way of assigning each variable a value in $\mathbb{R}$ so that all m linear constraints are satisfied, NO otherwise.

Let Integer Linear Programming be the same problem, but where each variable is only allowed to take values in $\mathbb{Z}$ rather than in $\mathbb{R}$.

(a) (11 points) Prove that Integer Linear Programming is in NP.

(b) (22 points) Prove that Integer Linear Programming is NP-hard.