601.433/633 Introduction to Algorithms **Fall 2025**
Homework #6 **Due:** Wednesday Dec 3, 2025, 11:59pm

Remember: you may work in groups of up to three people, but must write up your solution entirely on your own. 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 (40 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 k courses from subset S". However, any given course cannot be used towards satisfying multiple requirements. For example, if one requirement says that you must take at least two courses from $\{A, B, C\}$, and a second requirement states that you must take at least two courses from $\{C, D, E\}$, then a student who has taken just $\{B, C, D\}$ would not yet be able to graduate as C can only be used towards one of the requirements.

Your job is to give an efficient algorithm for the following problem: given a list of requirements $r_1, r_2, \ldots, r_m$ (where each requirement r_i is of the form "you must take at least k_i courses from set S_i"), and given a list L of courses taken by some student, determine if that student can graduate.

(a) (13 points) Given the m requirements and list of L courses taken (as above), design a flow network so that the maximum flow is $\sum_{i=1}^{m} k_i$ if and only if the student can graduate (and prove this).

(b) (13 points) Using the previous part, design an algorithm for the problem which runs in $O(|L|^2 m)$ time. Prove correctness and running time.

Now suppose that John Hopskins University changes their graduation requirements. Every time a student takes a class, they get some grade in $[0, 1]$. They must satisfy several requirements of the form "the sum of your grades in courses taken from set S must be at least k_i". The goal of this problem is to take on the role of the student, and figure out the least possible work they can do while still passing.

More formally, there is a set of n classes. Without loss of generality, we will simply say that this is the set $[n] = \{1, 2, \ldots, n\}$. We are also given m subsets $S_1, S_2, \ldots, S_m$ where each $S_j \subseteq [n]$, and m values $k_1, k_2, \ldots, k_m \in \mathbb{R}$. If a student puts in $x_i \in [0, 1]$ amount of work into class i, we assume that they will get x_i as a grade (i.e., the grade they receive is exactly equal to the amount of work they put into it). In order to graduate, for every $j \in [m]$, the sum of the grades they receive in the classes in S_j must be at least k_j (not taking a class is equivalent to putting in no work, and hence getting a grade of 0). Our goal is to minimize the total amount of work the student has to do while still graduating.

Note that unlike the previous requirements, now if some class i appears in both S_j and $S_{j'}$, then it will count towards both requirement j and requirement j'.

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

(c) (14 points) Show how this problem can be solved in polynomial time by using linear programming. Be sure to specify what the variables are, what the constraints are, and what the objective function is.

2 Graduation Requirements Revisited (35 points)

John Hopskins has switched to a more lenient policy for graduation requirements than in the last problem. As in the previous problem, there is a list of requirements $r_1, r_2, \ldots, r_m$ where each requirement r_i is of the form "you must take at least k_i courses from set S_i". However, under the new policy a student *may* use the same course to fulfill multiple requirements. 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. You will prove that the problem faced by this freshman is NP-complete, even if each k_i is equal to 1. 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) (10 points) Prove that this problem is in NP.

(b) (25 points)Prove that this problem is NP-hard.

3 Integer Linear Programming (25 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 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) (10 points) Prove that Integer Linear Programming is in NP.

(b) (15 points) Express the minimization problem from Question 2 as an integer linear program. Conclude that, unlike linear programming, integer linear programming is NP-hard.