Your homework submissions need to be typeset (hand-drawn figures are OK). See the course web page for suggestions on typing formulas.

The solution to each question need to be uploaded to CMS as a separate pdf file. To help provide anonymity in your grading, do not write your name on the homework (CMS will know it's your submission). (Questions with multiple parts need to be uploaded as a single file.)

For a proof that a problem is NP-complete, you must prove that the problem is both in NP and NP-hard. The proof of the first may often be very short (a few sentences) to motivate the certificate and certifier. The NP-hardness proof requires three parts: a reduction, an argument this reduction takes polynomial time in the size of the problem, and the proof of correctness of the reduction.

(1) Simplified Bloons Tower Defense

Inspired by a student's Piazza question, we will think about the hardness of solving Bloons Tower Defense. In this game, you are in charge of a team of N monkeys armed with weapons they can use to pop "bloons", or balloons, as they appear on a path on the map. The goal is to prevent the balloons from reaching the end of that path, which requires strategically placing monkeys to do damage effectively to oncoming balloons. Here, we remove time from the problem: every balloon is fixed in place, and each monkey can distribute some total amount of damage to any balloons in range. Your goal is to see if it is possible to choose a set of M monkeys from your team and place them to successfully pop all of the balloons. This problem is **NP**-Complete!

In this simplified Bloons Tower Defense, you are given:

- N monkeys, where the mth monkey has some integer total damage output d_m
- B stationary balloons, where each balloon b_i has some integer total health h_i
- L locations, where each location ℓ has a set of balloons A_ℓ that are within its range. A monkey at this location can only target the balloons within the location's range.

You would like to place as few monkeys as possible throughout these locations and direct each of those monkeys you placed to target a set of balloons within its range at that location. You do not need to use all L locations. A balloon b_i is considered popped if the sum of the damage directed at b_i from the monkeys targeting b_i equals or exceeds its health h_i. Each monkey is allowed to distribute its total damage output among its target balloons in any way as long as it inflicts integer non-negative damage to each target balloon. Let $d_{m,i}$ be the damage done by monkey m to balloon b_i.

The BloonsTowerDefense problem asks: **does there exist a valid assignment of at most M monkeys to these locations and balloons such that all the balloons can be popped?**

Show that BloonsTowerDefense is **NP**-Complete.

(2) Monotone Almost-All-True SAT

A monotone SAT formula is a SAT formula with no negated variables. So, for example,

$$\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_4)$$

is a monotone formula: each of x_1, x_2, x_3, x_4 only appears as a positive literal, never with a negation. A monotone formula is easy to satisfy: we can simply set all variables to be `true`, and this way all variables in each clause become true. Since we only need one variable to be `true` in each clause to satisfy the formula, this valuation easily satisfies the formula.

In this problem we have a stronger goal than typical satisfiability: we want to ensure that in each monotone SAT clause, at most one variable evaluates to false (we assume each clause has at least 2 variables). This is still easy to do by setting all variables to `true`, but becomes nontrivial if we also require our assignment to set some number of variables to `false`.

In the MONOTONE ALMOST-ALL-TRUE SAT problem, we are given a monotone formula Φ and an integer k. We ask if there is a way to set k variables to `false` (and the rest to `true`) such that no more than 1 variable in each clause is `false`. For example, for the formula above, if $k = 1$, then we can find an assignment satisfying these constraints by setting any one variable to `false`. It is impossible, however, to find an assignment that sets $k = 2$ variables to `false` while ensuring that at most one variable in each clause is `false`.

Show that the MONOTONE ALMOST-ALL-TRUE SAT problem is NP-complete.