- **Do not use external sources** beyond the materials linked to on the course website to solve these problems.
- You are encouraged to work with (≤ 2) other students, but you must **write your solutions independently**.
- Be sure to **list your collaborators** by name clearly at top of your submission, or "no collaborators" if none.

Classification: Complexity (S24)

For the following problems, figure out whether it is in P or NP-complete, and prove it. For P this means giving a poly-time algorithm. For NP-complete, this means three things: (1) showing it's in NP, (2) giving a reduction from an NP-complete problem we cover in class, and (3) showing that reduction runs in polynomial time.

In addition to the problems we cover in class, there are a couple others in §12.14 of the following which might be handy on PS11 and/or PS12: `https://jeffe.cs.illinois.edu/teaching/algorithms/book/12-nphard.pdf`. (You may recognize some of the figures from my slides – they were all by Jeff Erickson. I'd encourage you to read the whole chapter; he does a phenomenal job.) In particular, note SubsetSum, Partition, & 3Partition, which are of a different variety than the problems we've discussed.

1. You are trying to design a class schedule so that students have no conflicts for the courses they need to take. Given a set of students S, a set of courses C, a set of course requests $\{R_s \subseteq C : s \in S\}$ from students, and the number of time slots k, you need to decide whether there is a schedule $h : C \to \{1, \dots, k\}$ mapping courses to time slots such that no student has a scheduling conflict. In other words, a valid schedule h is one for which, for all students $s \in S$, if $c, c' \in R_s$ then $h(c) \neq h(c')$.

This problem is in NP, as checking the correctness of a candidate solution takes a maximum time of $O(|C| * |S|)$ time (checking that a given student has no overlapping classes takes a maximum time of $O(|C|)$, and this must be done for each student).

We will be reducing from the K-COLORING problem.

First, we will set a limitation on the described scheduling problem such that every student requests exactly two courses.

Each node in the K-COLORING problem will be represented as a course in the limited scheduling problem. Each edge in the K-COLORING problem will be represented as a request in the limited scheduling problem (since edges connect two nodes, and requests essentially connect two courses). The number of colors k in the K-COLORING problem will be represented by the number of time slots in the limited scheduling problem.

So, if this instance of the scheduling problem has a satisfactory solution, so too does the original instance of the K-COLORING problem (as a request with any overlapping courses means that two same-colored nodes are connected; so if there are no requests with overlapping courses, there are no two connected nodes that have the same color). Therefore this is a valid reduction.

This reduction runs in no greater than polynomial time, as follows:

- Representing each node as a course takes $O(|C|)$ time.
- Representing each edge as a request takes $O(2 * |R_s|)$ time (since each request has two courses).
- Representing the number of colors as the number of time slots takes $O(1)$ time.

Therefore, the described scheduling problem is NP-complete.

2. An astronaut is gathering n rock samples, and needs to fit them into 4 cases. Each case can carry up to W kilograms. Given n, W, and the weight $w_i \in \mathbb{N}$ in kilograms of each rock $i \in \{1, \ldots, n\}$, she must decide whether she can pack all rocks without going over the weight limit.

This problem is in NP, as checking the correctness of a candidate solution takes $O(n + 4) = O(n)$ time (checking that each rock is in a case $= O(n)$, checking that each case is not overfilled $= O(4)$).

We will be reducing from the PARTITION problem as explained on page 405 of the above University of Illinois link.

Let Q be the sum of the integers in the set S. To the original set S, append two new integers, each of value $\frac{Q}{2}$.
The integers in S represent the respective weight w_i of each rock i in the described problem. Each case would have a capacity of $W = \frac{Q}{2}$. The two rocks of weight $\frac{Q}{2}$ would be put into the third and fourth cases, filling them completely. Then, the rocks with weights of the original integers of S would have to fit into the first two cases in order to confirm the correctness of a candidate solution. So, if this instance of the rock problem has a satisfactory solution, so too does the original instance of the PARTITION problem. Therefore this is a valid reduction.

This reduction runs in no greater than polynomial time, as adding two new integers to the original set S and placing the two rocks weight $\frac{Q}{2}$ into the third and forth cases all runs in constant time.

Therefore, the described rock problem is NP-complete.