

Reductions: Written Problems

1. In class, we learned about the complexity classes P , NP , and others. There is another complexity called $PSPACE$, which is the set of problems that can be solved using a polynomial amount of space (with a deterministic turing machine). What is the relationship between NP and $PSPACE$? Is one a subset of the other? Are they different sets? Prove your answer.
2. The slides present (but do not solve) a problem called *Lecture Planning*. For this question, you will prove that the *Lecture Planning* problem is NP-Complete. The problem is re-stated here for your convenience:

Suppose for a course, that there are n total (unique) guest lecturers that can visit the class to present. One guest lecture will be given for each of the first l weeks of the course. For each of these weeks, you are given a subset of the n guest lecturers who are available that week (notice that most of these lecturers will have multiple free weeks).

During the second half of the course, there are p projects that must be completed. For each project, there is a list of guest lectures that will allow the students to be able to complete that project successfully (These lists might overlap, i.e., one guest lecture might allow multiple projects to be completed). The problem, thus, is to choose a schedule of l guest lecturers, one per week, that will ensure the students can successfully complete each project.

Prove that this problem is in NP . Describe the verification algorithm.

3. Prove that lecture-planning is in NP -HARD by showing that $3\text{-SAT} \leq_p LP$ where LP is the lecture-planning problem. Prove that the outputs of the two problems is always the same (notice this is a bi-conditional).
4. For this problem, we will prove that the *knapsack problem* from class is NP-Complete. First, consider the *ethical knapsack problem* (which is NP-Complete) and states the following: Given an array of the value of items a that can be stolen, a thief wants to steal a combination of items such that the total value is exactly some sum s (the thief is "ethical" in the sense that they only want to steal enough to pay off their debts, but no more. Also, the thief has no restriction on how much they can carry). Any algorithm would return yes or no: whether or not it is possible to steal the exact value s worth of items.

Also, recall the *knapsack problem* from class in which a thief has a knapsack capacity C and the items have corresponding values and weights. Show the following:

- formulate the decision problem for the *knapsack problem* (from class).
- show that the *knapsack problem* is in NP by providing a verification algorithm.
- show that the *knapsack problem* is NP-Hard by producing a reduction from the *ethical knapsack problem*.