1  NP-hardness I

3SAT is a version SAT where each clause has exactly 3 variables. Explain what is wrong with the following “proof” that 3SAT is NP-hard: Given an instance of 3SAT, we can use a SAT solver to determine if a satisfying assignment exists. This provides a polynomial time transformation that gives an answer to 3SAT instances.

2  NP-hardness II

Prove that the following decision problem is NP-hard: Given a SAT instance and an integer k, does there exist a satisfying assignment in which at least k variables take the value true?

3  Approximation I

Provide a greedy approximation algorithm for the node cover problem and bound its suboptimality.

4  Approximation II

Consider a version of set cover where instead of choosing sets from one big set $C$, you must choose sets from $C_1 \ldots C_k$. The decision problem now asks whether it’s possible to cover all of the atoms using one element from each $C_i$, $1 \leq i \leq k$. First, show that this problem is NP-hard, then explain why the submodular maximization trick doesn’t apply here.

5  Planning

Reduce 3-coloring to planning, where planning uses PDDL as the input language. Note that it’s significantly trickier to get the details right on this than the other hardness reductions you’ve seen so far. You’ll need to put some thought into this one.