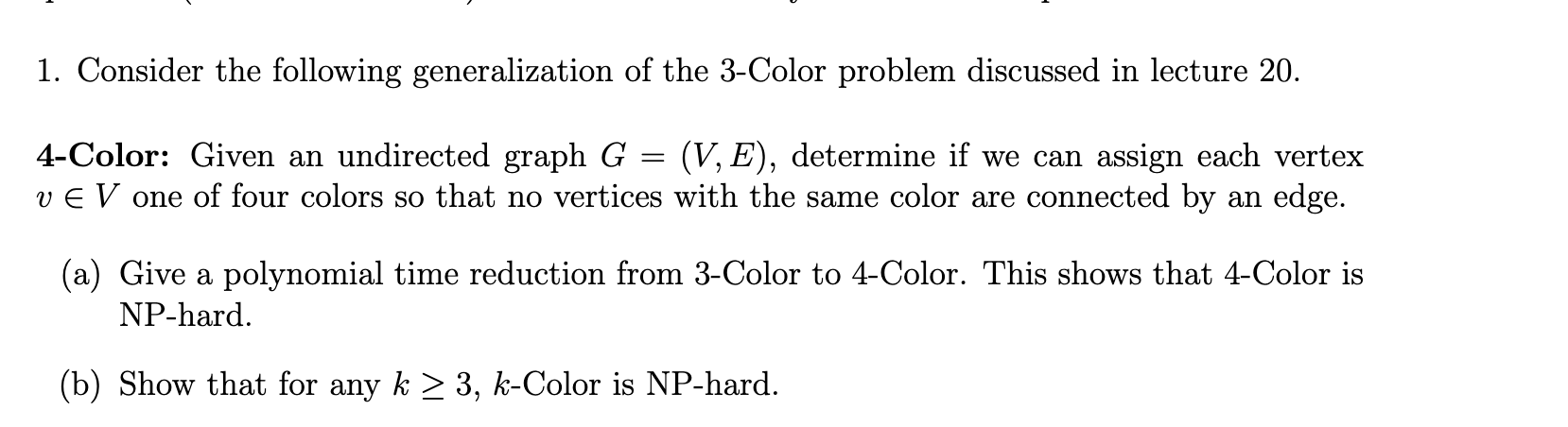
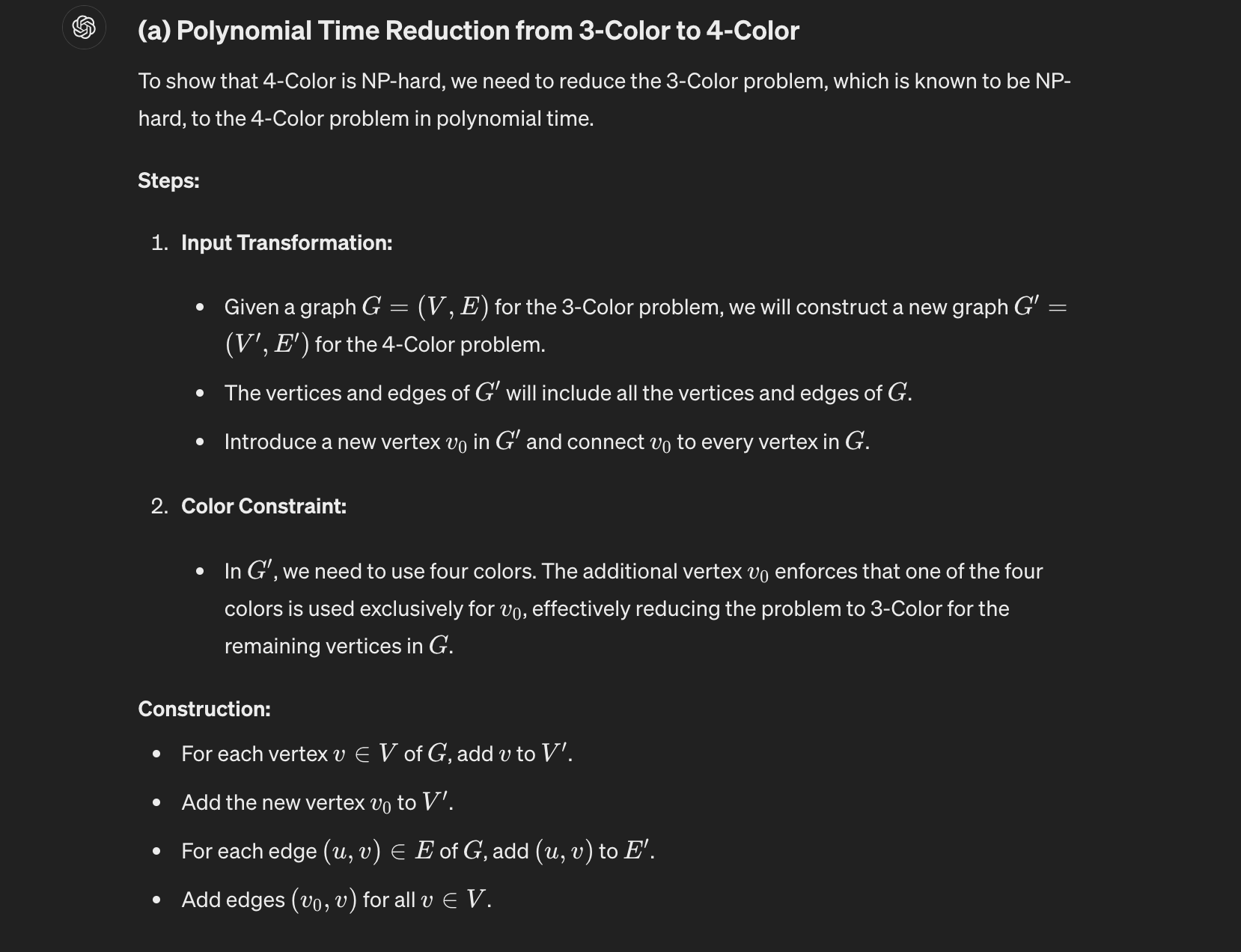
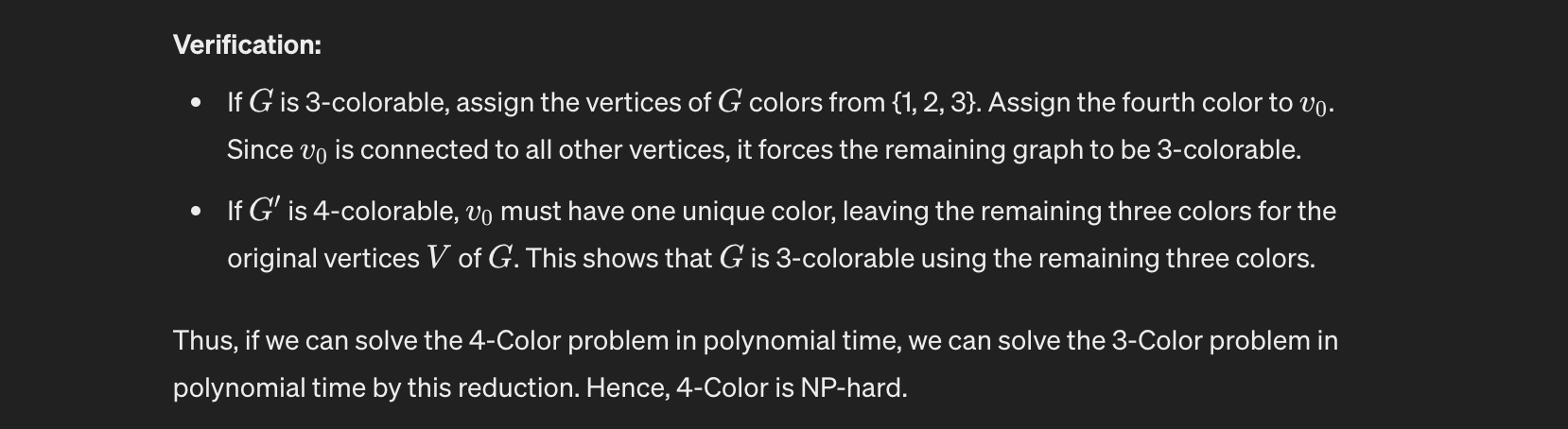
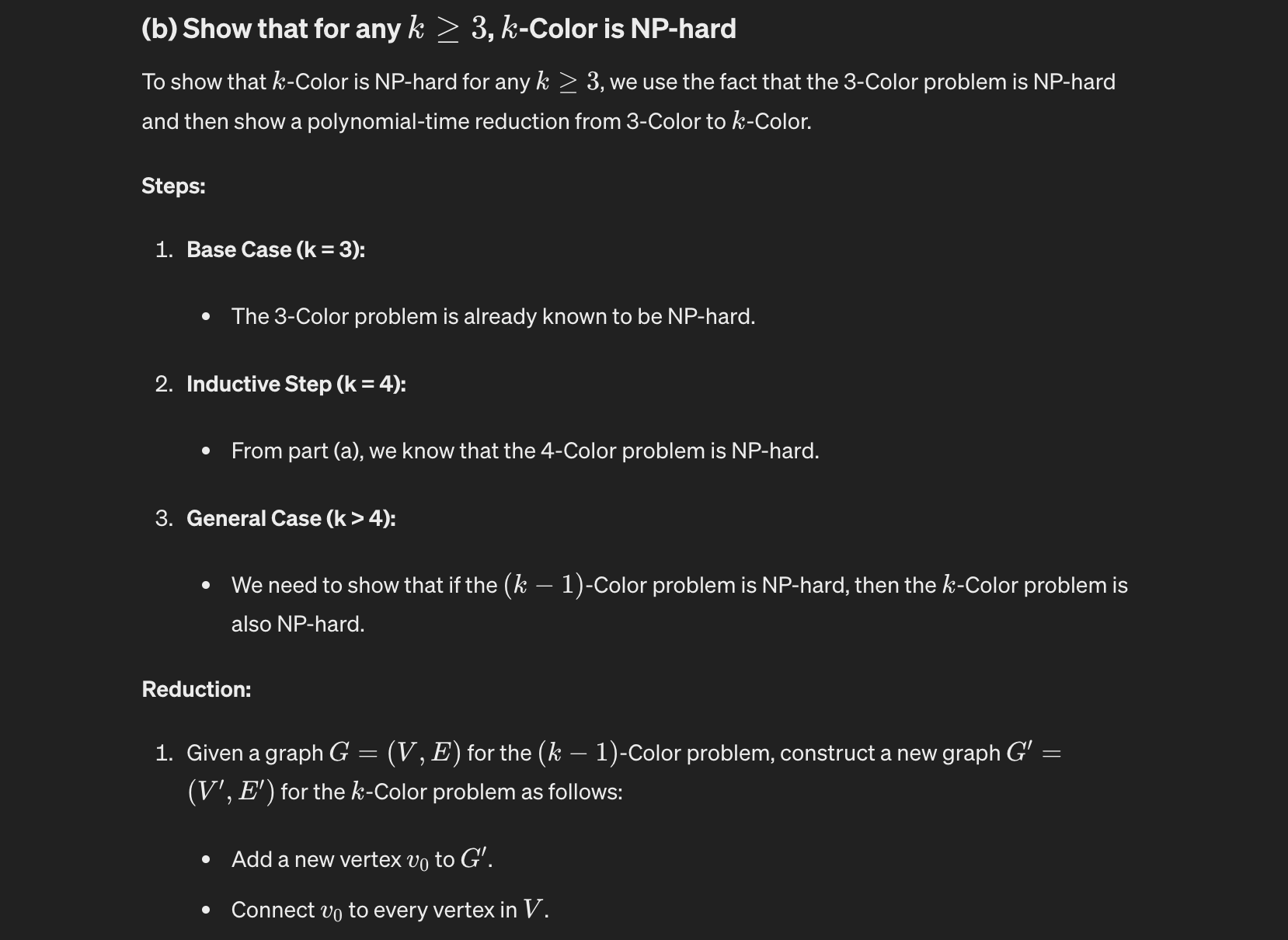
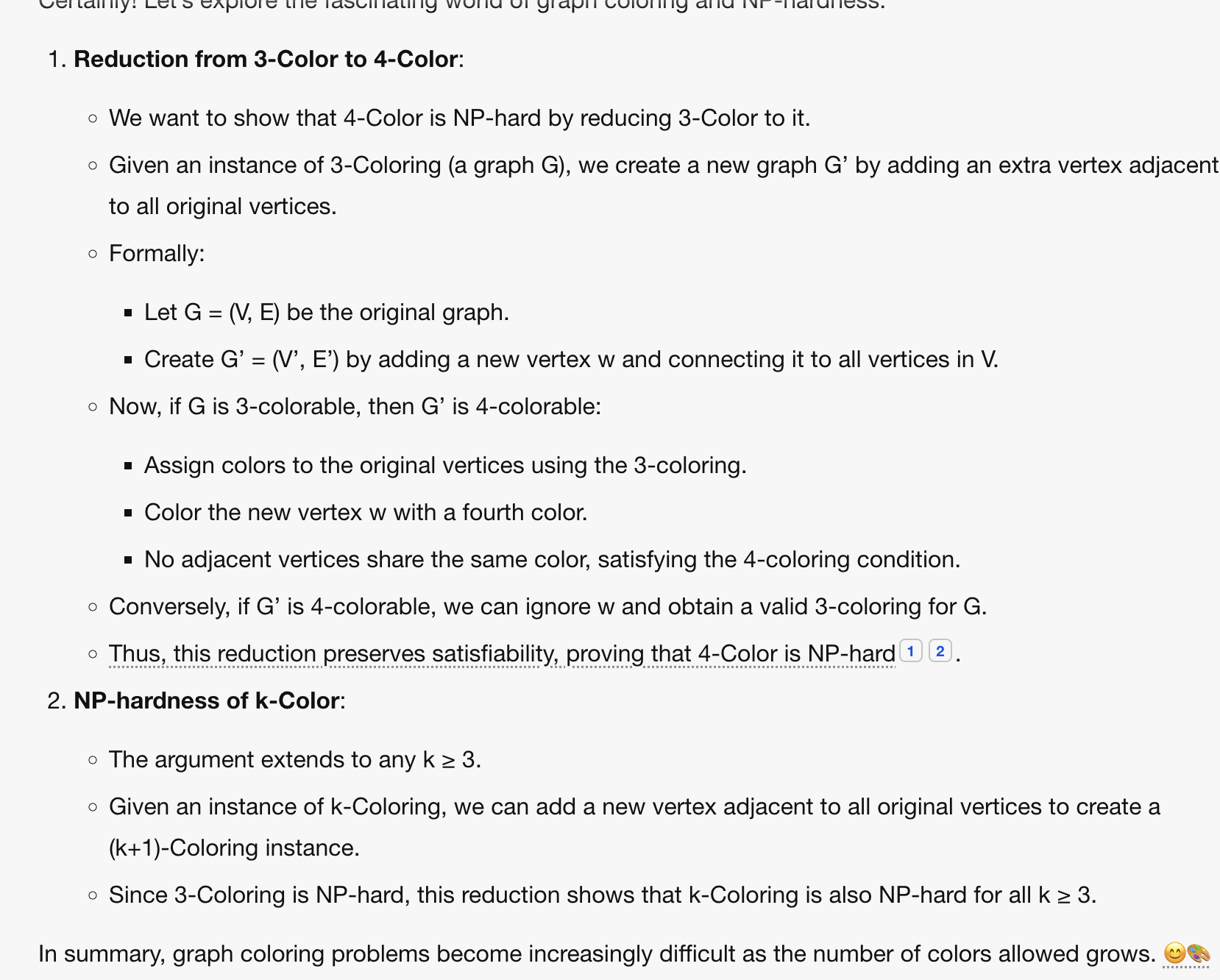
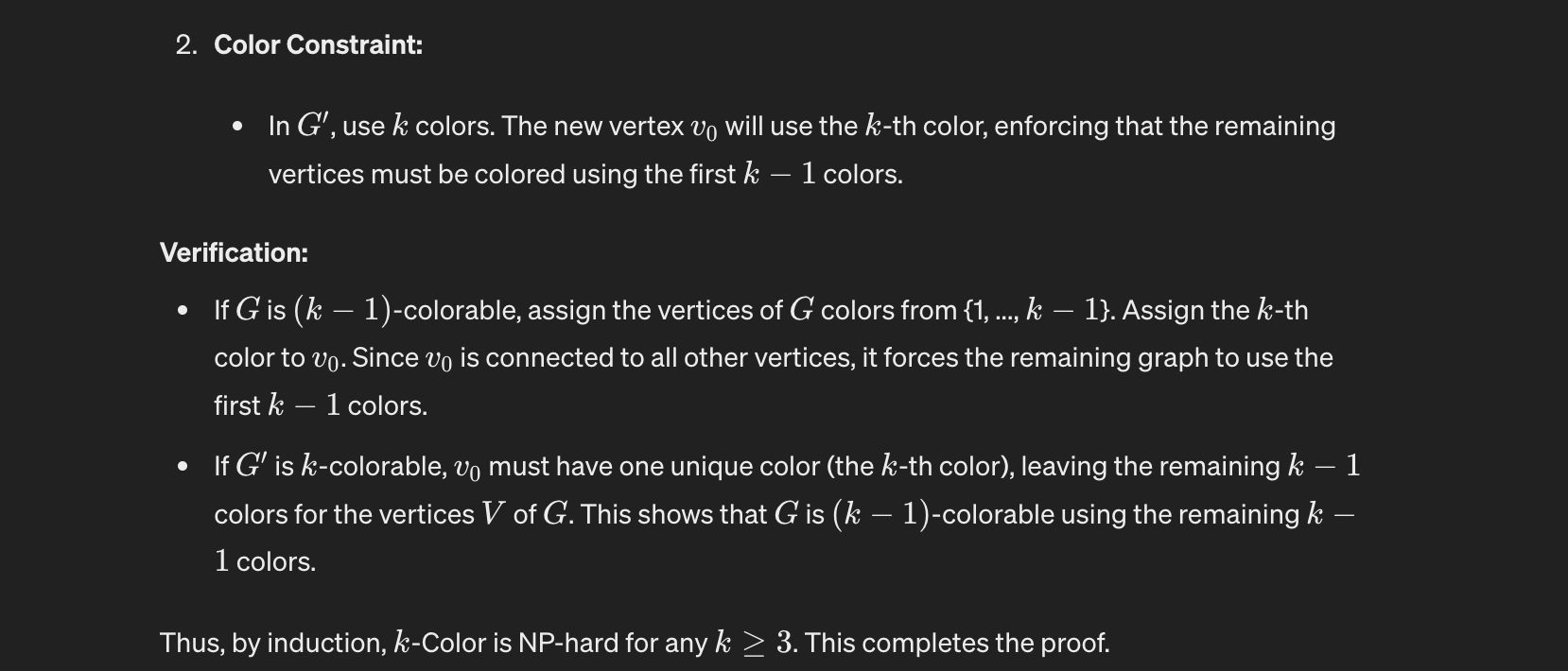
## Problem 1



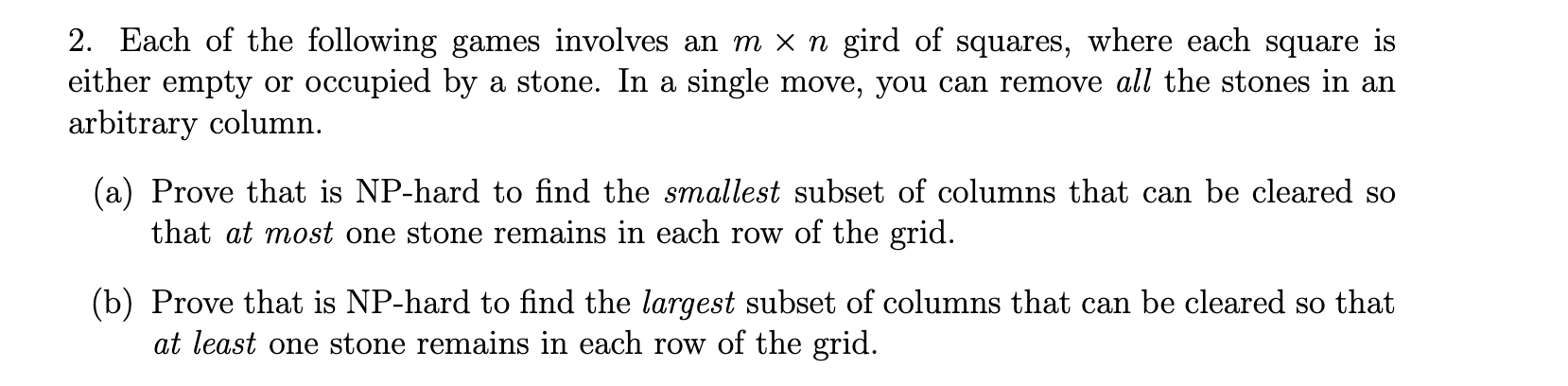


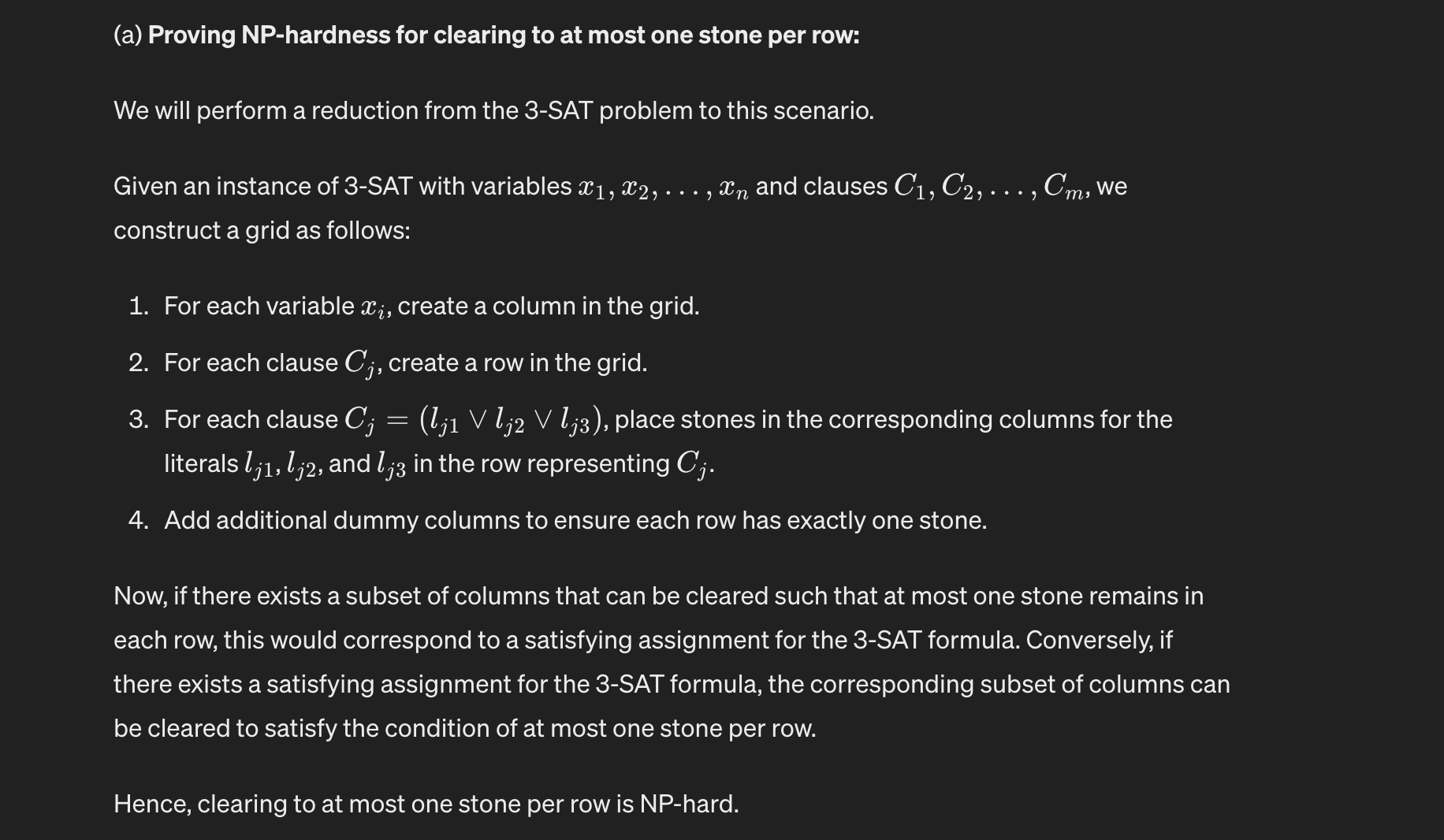


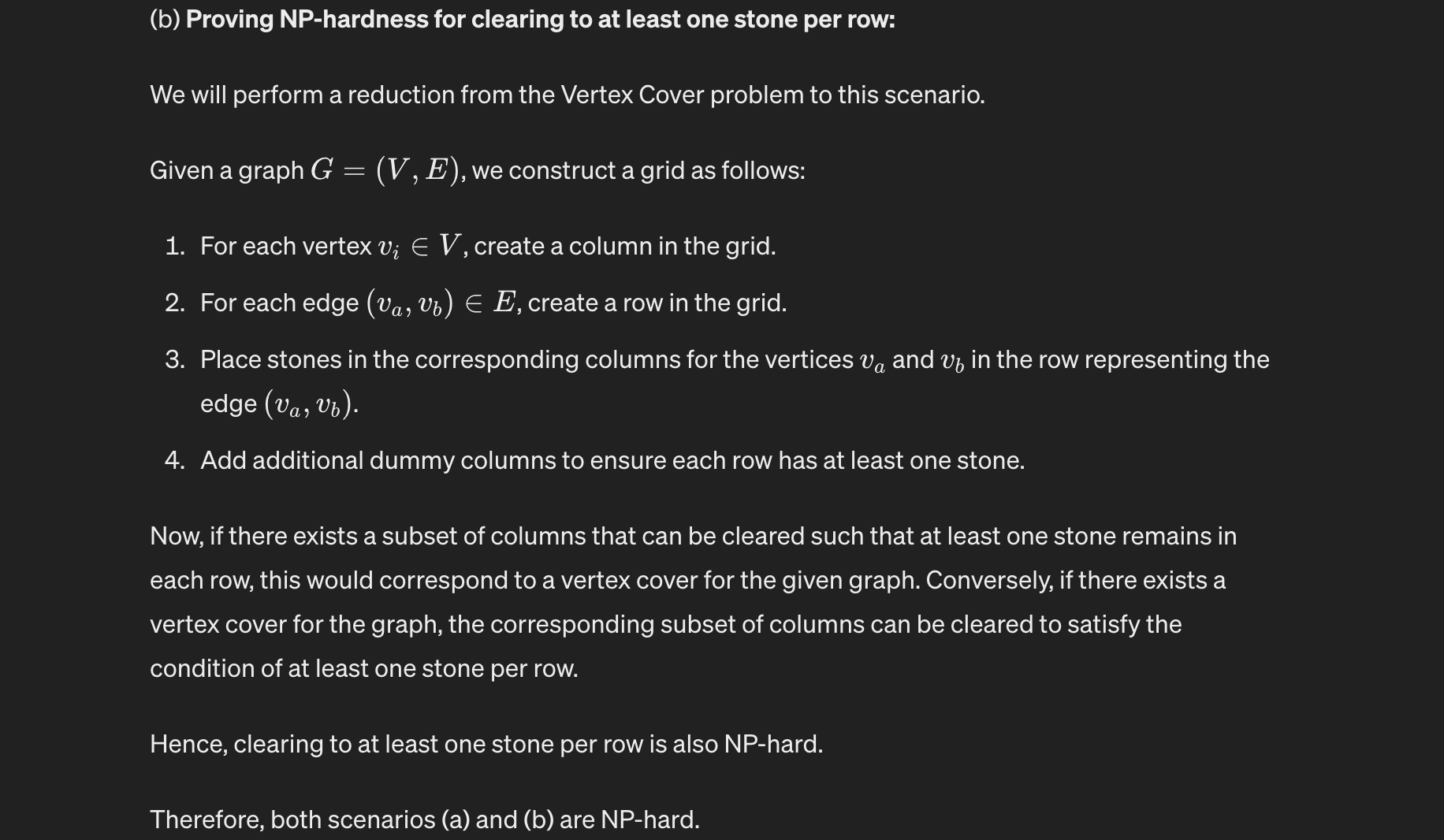




## Problem 2







Certainly! Let’s delve into these intriguing problems.

**(a) NP-Hardness for the Smallest Subset:** To prove that finding the smallest subset of columns is NP-hard, we’ll reduce from the Exact Cover by 3-Sets (X3C) problem. In X3C, we’re given a set of elements and a collection of 3-element subsets, and the goal is to find a subset of these subsets that covers all elements exactly once.

1. **Problem Statement:**
   * We have an n × m grid of squares.
   * Each square can be empty or occupied by a stone.
   * The objective is to remove columns such that at most one stone remains in each row.
2. **Reduction:**
   * Given an instance of X3C, construct the following grid:
     + Rows represent the elements.
     + Columns correspond to the 3-element subsets.
     + Place a stone in a cell if the corresponding element is part of the subset.
   * Let S be the set of columns with stones.
3. **Claim:**
   * If there exists an exact cover for X3C, then there exists a subset of columns S’ (with at most one stone per row) in our grid.
   * Conversely, if we find such an S’, we can construct an exact cover for X3C.
4. **Proof:**
   * If X3C has an exact cover, then S’ corresponds to the selected subsets.
   * Conversely, if we have S’, it covers all elements exactly once (since each row has at most one stone).
   * [Thus, the smallest subset of columns that can be cleared so that at most one stone remains in each row of the grid must correspond to the columns containing the stones corresponding to S’1](https://www.cliffsnotes.com/tutors-problems/Computer-Science/47298436-Each-of-the-following-games-involves-an-n-m-grid-of-squares-where/).

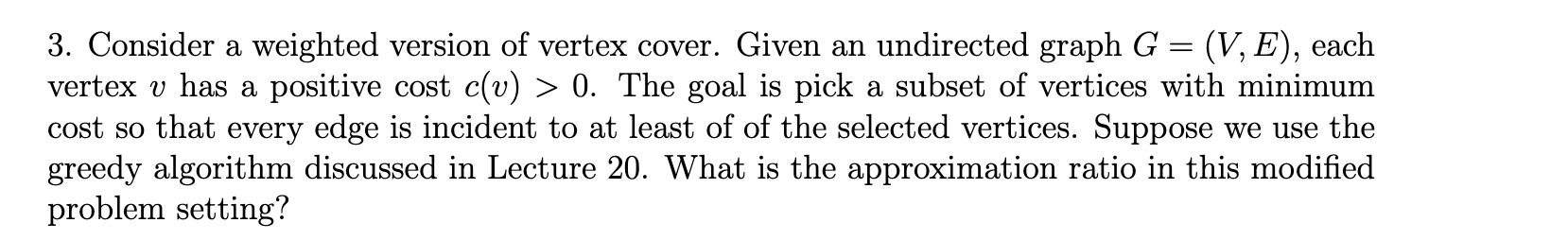
**(b) NP-Hardness for the Largest Subset:** To prove that finding the largest subset of columns is NP-hard, we’ll reduce from the Vertex Cover problem.

1. **Problem Statement:**
   * We have the same n × m grid as before.
   * The goal now is to remove columns such that at least one stone remains in each row.
2. **Reduction:**
   * Given an instance of Vertex Cover with a graph G, construct the following grid:
     + Rows represent the vertices of G.
     + Columns correspond to the edges of G.
     + Place a stone in a cell if the corresponding vertex is an endpoint of the edge.
   * Let S be the set of columns with stones.
3. **Claim:**
   * If G has a vertex cover of size k, then there exists a subset of columns S’ (with at least one stone per row) in our grid.
   * Conversely, if we find such an S’, it corresponds to a vertex cover of G.
4. **Proof:**
   * If G has a vertex cover, then S’ corresponds to the selected edges.
   * Conversely, if we have S’, each row has at least one stone (covering the vertices), forming a vertex cover.
   * [Thus, the largest subset of columns that can be cleared so that at least one stone remains in each row of the grid corresponds to the columns containing the stones corresponding to S’1](https://www.cliffsnotes.com/tutors-problems/Computer-Science/47298436-Each-of-the-following-games-involves-an-n-m-grid-of-squares-where/).

In summary, both problems are NP-hard, and their reductions demonstrate the hardness of finding the smallest and largest subsets of columns satisfying the given conditions. 🧩🔍🌟

## 

## Problem 3



1. If simple greedy cover where we pick both vertices per edge then approx = 2 as OPT will pick 1 vertex
2. If greedycover2 where we pick a vertex with min weight/ # newly covered edges we get approx as Hn as OPT will pick mid of star graph where weight = 1+e while greedy will pick 1/1 + ½ + ⅓ + ¼ weighted vertices. ALG/OPT <=Hn
3. If we use primal dual cover where we pick partial vertex, then we get approx as 2

