Problem 7.31. Consider the following scheduling problem. You are given a list of final exams $F_1, \ldots, F_k$ to be scheduled, and a list of students $S_1, \ldots, S_l$. Each student is taking some specified subset of these exams. You must schedule these exams into slots so that no student is required to take two exams in the same slot. The problem is to determine if such a schedule exists that uses only h slots. Formulate this problem as a language and show that this language is NP-complete.

Proof. Let $SCHEDULE = \{\langle F, S, T, h\rangle \mid F = \{F_1, \ldots, F_k\}$ is a set of final exams, $S = \{S_1, \ldots, S_l\}$ is a set of students, $T \subseteq S \times F$ is set of ordered pairs (S_i, F_j), where student S_i takes exam F_j, and a schedule of h slots exists so that no student is required to take two exams in the same slot$\}$.

To show that $SCHEDULE$ is NP-complete, we must show that it is in NP and that all NP-problems are polynomial time reducible to it. The first part is easy; a certificate is simply the schedule. To prove the second part, we show that $3COLOR$ is polynomial time reducible to $SCHEDULE$. The reduction converts an undirected graph G into a set of final exams F, a set of students S, a set of ordered pairs (S_i, F_j), where student S_i takes exam F_j, and an integer h, so that G is colorable with 3 colors, iff a schedule of h slots exists so that no student is required to take two exams in the same slot.

Let G be any undirected graph with s nodes. Now we show how to convert G to sets F, S, T and integer h. We choose h to be 2, and construct sets F, S and T by partitioning the nodes of G into exam or student nodes as follows:

1. Repeat for each node $n_i = n_1, n_2, \ldots, n_s$.
2.   If $n_i \notin S$ and $n_i \notin F$, then
3.   If all adjacent nodes $m_1, m_2, \ldots m_r$ of node n_i are mutually connected, then
4.   Add n_i in S.
5.   Repeat for each node $m_j = m_1, m_2, \ldots m_r$.
6.   Add node m_j in F.
7.   Add ordered pair (n_i, m_j) in T.
8.   Otherwise, add n_i in F.

Suppose that graph G is colorable with 3 colors. Then, there are three cases:

1. All nodes are student nodes.
2. All nodes are exam nodes.
3. There is a mix of students and exam nodes.

In cases 1 and 2, the required 2 slot schedule can be constructed trivially by placing all exams in one of the slots. In the third case, if all the nodes in the set of students S are assigned the same color, then the nodes in the set of exams F can only be assigned two different colors. Such a coloring gives the schedule with 2 slots so that no student is required to take two exams in the same slot. Next, we show how to reassign colors so that all student nodes have the same color.

1. Let c_1 be the first color.
2. Repeat for each student node $n_i \in S$.
3. Let c_2 be the color of n_i.
4. If $c_1 \neq c_2$, then
5. Scan all the exam nodes connected with the student node n_i. If an exam node has color c_1, then assign it c_2. There can be only one such exam node.
6. Assign color c_1 to student node n_i.

Suppose a 2 slot schedule exists for a set of students S and exams F. Then, the nodes of the graph G can be colored with 3 colors as follows:

1. Assign first color to all the student nodes.
2. Assign second color to all the exam nodes that are assigned first slot.
3. Assign third color to all the exam nodes that are assigned second slot.

□