1. Let

\[ \text{SetPACK} = \{ \langle C, k \rangle : C \text{ is a collection of finite sets, at least } k \text{ of which are pairwise disjoint} \}. \]

Show that \text{SetPACK} is NP-complete.
Hint: reduction from \text{Independent Set}.

2. You are given a box and a collection of cards as indicated in the following figure. Because of the pegs in the box and the notches in the cards, each card will fit in the box in either of two ways. Each card contains two columns of holes, some of which may not be punched out. The puzzle is solved by placing all the cards in the box so as to completely cover the bottom of the box (i.e., every hole position is blocked by at least one card that has no hole there). Let

\[ \text{PUZZLE} = \{ \langle c_1, \ldots, c_k \rangle : \text{the puzzle with cards } c_1, \ldots, c_k \text{ has a solution} \}. \]

Show that \text{PUZZLE} is NP-complete.
Hint: reduction from \text{3SAT}.

3. Show that if \( P = NP \), a polynomial time algorithm exists that produces a satisfying assignment when given a satisfiable Boolean formula.

Note: the algorithm you are asked to provide computes a function, but \( NP \) contains languages, not functions. The \( P = NP \) assumption implies that \( SAT \in P \), so there is a polynomial-time algorithm \( A \) that tests whether a given Boolean formula is satisfiable — but we do not know how \( A \) works, and \( A \) need not reveal any satisfying assignments. Your task is to design a polynomial-time algorithm \( B \) for computing a satisfying assignment — algorithm \( B \) may make \textit{many} calls to \( A \), with various inputs.

4. Show that the \textit{SUBSET-SUM} problem can be solved in polynomial time provided the numbers \( d_1, \ldots, d_n, t \) are bounded by a polynomial in \( n \). In particular, assuming that \( 0 \leq d_1, \ldots, d_n, t \leq B \), give an algorithm based on dynamic programming that determines if there is a subset \( I \subseteq \{1, \ldots, n\} \) such that \( \sum_{i \in I} d_i = t \) that runs in time bounded by a polynomial in \( n \) and \( B \).

5. For positive integer \( k \), let \( kRSAT \) be the set of encodings \( \langle \phi \rangle \), such that \( \phi \) is a satisfiable Boolean formula in conjunctive normal form, where each variable occurs in at most \( k \) distinct clauses. Show that \( 3RSAT \) is NP-complete.

Hint: reduction from \textit{3SAT}. You may make use of the following fact (discussed in class): every Boolean formula \( \psi \) in 1–3 variables may be rewritten as an equivalent 3-CNF formula \( N(\psi) \).
6. **Honover’s exercise: Resolution.** This problem develops the foundations underlying a basic technique, called *resolution*, that is used in automated theorem proving, along with some algorithmic applications.

(a) Let \(A_1, \ldots, A_m, B_1, \ldots, B_n,\) and \(C\) be arbitrary Boolean formulas, none of which contain the variable \(x\). Show that the formula

\[
\left( \bigwedge_{i=1}^{m} (x \lor A_i) \right) \land \left( \bigwedge_{j=1}^{n} (\bar{x} \lor B_j) \right) \land C
\]

is satisfiable if and only if the formula

\[
\left( \bigwedge_{1 \leq i \leq m} (A_i \lor B_j) \right) \land C
\]

is satisfiable.

(b) Using part (a), show that \(2RSAT\) (see previous exercise) is in \(P\).

(c) Let \(2SAT\) be the set of encodings \(\langle \phi \rangle\), such that \(\phi\) is a satisfiable Boolean formula in conjunctive normal form, where each clause consists of two distinct literals. Using part (a), show that \(2SAT\) in \(P\).