1. Are the below Boolean formulas satisfiable?
   (a) \((a \lor c) \land (\bar{a} \lor b) \land (a \lor \bar{c}) \land (\bar{a} \lor \bar{b} \lor c) \land (\bar{a} \lor \bar{b} \lor \bar{c})\)
   (b) \((a \lor b \lor c) \land (\bar{a} \lor d \lor f) \land (\bar{b} \lor e \lor \bar{f}) \land (\bar{c} \lor \bar{d} \lor \bar{e})\)

   Either give some satisfying truth assignment, or argue that there is no such.

2. Present an algorithm that returns the truth value of a Boolean formula, when the formula is given as an expression tree and the truth assignment in a suitable form, say, in an array. Analyze the time complexity of the algorithm. (If you do not know what an “expression tree” is, try googling the Net for it on.)

3. Sketch a deterministic algorithm that tests whether a given Boolean formula is satisfiable or not. (Use the solution of the above assignment as a subroutine.) Analyze the time complexity. Based on this, explain that the satisfiability of a Boolean formula can be tested in polynomial time if the number of its variables is bounded by a constant.

4. We say that a Boolean formula is \textit{refutable}, if some truth assignment makes it \textit{false}. Let’s define the language
   \[
   \text{REFUTABLE} = \{w \mid w \text{ is a refutable Boolean formula}\}.
   \]

   Show that \text{REFUTABLE} is NP-complete. (Hint: an \textit{easy} polynomial-time reduction.)

5. An instance of the \textit{Partition} problem consists of a set of items \(A = \{a_1, \ldots, a_m\}\), each with a weight \(w(a_i) \in \mathbb{N}\). The task is to test whether the set can be divided in two equally heavy subsets \(A' \subseteq A\) and \(A - A'\), that is,
   \[
   \sum_{a \in A'} w(a) = \sum_{a \in A - A'} w(a).
   \]

   \textit{Partition} is known to be NP-complete. Based on this, give an alternative proof for the NP-completeness of the (decision version of the unweighted) Knapsack problem discussed at the lecture.