P and NP

1. Suppose $\mathsf{P} \neq \mathsf{NP}$. Could there exist a polynomial-time algorithm for translating a CNF into an equivalent DNF?

2. *Reducibility.* Recall that, for two decision problems, $A \leq_m^P B$ if there exists a polynomial time computable function f such that $x \in A \iff f(x) \in B$, for any x.
   Consider the following decision problems:

   - INDSET: given a graph G and a number k, decide whether G contains an independent set of k vertices (that is, k vertices, none of which are connected);
   - CLIQUE: given a graph G and a number k, decide whether G contains a clique of k vertices (that is, k vertices, which are connected pairwise);
   - VERTEXCOVER: given a graph G and a number k, decide whether G has a vertex cover of k vertices (that is, a set U of k vertices such that for every edge at least one end of belongs to U).

   Show that: (a) INDSET $\leq_m^P$ CLIQUE and CLIQUE $\leq_m^P$ INDSET; (b) INDSET $\leq_m^P$ VERTEXCOVER.

3. Show that if $\mathsf{NP} \neq \mathsf{coNP}$, then $\mathsf{P} \neq \mathsf{NP}$.

4. (a) Suppose SAT $\in \mathsf{P}$. Show that there exists a polynomial time algorithm which checks satisfiability of Boolean formulae and, if a given formula is satisfiable, yields a satisfying assignment.
   (b) Does the same work for 2-SAT?

5. (a) Does there exist a polynomial time algorithm that, given a 2-CNF, yields *all* its satisfying assignments?
   (b) Does there exists an algorithm for generating all satisfying assignments of a given 2-CNF with *polynomial delay?* That means that the algorithm should produce the answers (satisfying assignments) gradually, one by one, spending a polynomially bounded amount of time before the first answer and between answers.