P and NP

1. Show that any problem in the NP class is decidable in exponential (i.e., $2^{\text{poly}(n)}$) time.
2. Prove that if an NP-hard problem belongs to P, then P = NP.
3. Suppose P ≠ NP. Could there exist a polynomial-time algorithm for translating a Boolean formula into an equivalent DNF?
4. Show that if NP ≠ coNP, then P ≠ NP.
5. (a) Suppose SAT ∈ 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 (checking satisfiability for 2-CNF)?
6. (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.
7. Suppose P ≠ NP. Could there exist a polynomial-time algorithm which, given a Boolean formula φ, answers whether the number of its satisfying assignments is greater or equal than 2?