Exercises III

Note that to *give an algorithm* means not only to describe the algorithm, but also to analyze its running time.

**III-1** Show that the subset-sum problem is solvable in polynomial time if the target value $t$ is expressed in unary.

**III-2** *(CLRS 34.1-5)* Show that if an algorithm makes at most a constant number of calls to polynomial-time subroutines and performs an additional amount of work that also takes polynomial time, then it runs in polynomial time. Also show that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

**III-3** *(CLRS 34.5-8)* In the half 3-CNF satisfiability problem, we are given a 3-CNF formula $\phi$ with $n$ variables and $m$ clauses, where $m$ is even. We wish to determine whether there exists a truth assignment to the variables of $\phi$ such that exactly half the clauses evaluate to 0 and exactly half the clauses evaluate to 1. Prove that the half 3-CNF satisfiability problem is NP-complete. (You may assume that the 3-CNF formula has at most 3 literals per clause, not necessarily exactly 3.)

**III-4** *(CLRS 34.4-6)* Suppose someone gives you a polynomial-time algorithm to decide formula satisfiability. Describe how to use this algorithm to find satisfying assignments in polynomial time.

**III-5** *(CLRS 34.5-6)* Show that the hamiltonian-path problem is NP-complete. (You may assume that you know that HAM-CYCLE is NP-complete.)