Exercises 34.1-5

Show that an otherwise polynomial-time algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

Exercises 34.2-3

Show that if HAM-CYCLE $\in$ P, then the problem of listing the vertices of a hamiltonian cycle, in order, is polynomial-time solvable.

Exercises 34.2-5

Show that any language in NP can be decided by an algorithm running in time $2^{O(n^k)}$ for some constant *k*.

Exercises 34.4-1

Consider the straightforward (nonpolynomial-time) reduction in the proof of Theorem 34.9. Describe a circuit of size *n* that, when converted to a formula by this method, yields a formula whose size is exponential in *n*.

Exercises 34.4-7

Let 2-CNF-SAT be the set of satisfiable boolean formulas in CNF with exactly 2 literals per clause. Show that 2-CNF-SAT $\in$ P. Make your algorithm as efficient as possible. (*Hint:* Observe that $x \vee y$ is equivalent to $\neg x \rightarrow y$. Reduce 2-CNF-SAT to a problem on a directed graph that is efficiently solvable.)

Exercises 34.5-5

The ***set-partition problem*** takes as input a set *S* of numbers. The question is whether the numbers can be partitioned into two sets *A* and $\bar{A} = S - A$ such that $\Sigma_{x \in \bar{A}} x$. Show that the set-partition problem is NP-complete.

Exercises 34.5-7

The ***longest-simple-cycle problem*** is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. Show that this problem is NP-complete.