

问题与反馈

2015.5.9

34.1-2

Give a formal definition for the problem of finding the longest simple cycle in an undirected graph. Give a related decision problem. Give the language corresponding to the decision problem.

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.

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.

34.2-6

A *hamiltonian path* in a graph is a simple path that visits every vertex exactly once. Show that the language $\text{HAM-PATH} = \{\langle G, u, v \rangle : \text{there is a hamiltonian path from } u \text{ to } v \text{ in graph } G\}$ belongs to NP.

The *complexity class* **NP** is the class of languages that can be verified by a polynomial-time algorithm.⁸ More precisely, a language L belongs to NP if and only if there exist a two-input polynomial-time algorithm A and a constant c such that

$$L = \{x \in \{0, 1\}^* : \text{there exists a certificate } y \text{ with } |y| = O(|x|^c) \\ \text{such that } A(x, y) = 1\}.$$

We say that algorithm A *verifies* language L *in polynomial time*.

34.2-11

Let G be a connected, undirected graph with at least 3 vertices, and let G^3 be the graph obtained by connecting all pairs of vertices that are connected by a path in G of length at most 3. Prove that G^3 is hamiltonian. (*Hint:* Construct a spanning tree for G , and use an inductive argument.)

34.4-3

Professor Jagger proposes to show that $\text{SAT} \leq_p \text{3-CNF-SAT}$ by using only the truth-table technique in the proof of Theorem 34.10, and not the other steps. That is, the professor proposes to take the boolean formula ϕ , form a truth table for its variables, derive from the truth table a formula in 3-DNF that is equivalent to $\neg\phi$, and then negate and apply DeMorgan's laws to produce a 3-CNF formula equivalent to ϕ . Show that this strategy does not yield a polynomial-time reduction.

34.4-5

Show that the problem of determining the satisfiability of boolean formulas in disjunctive normal form is polynomial-time solvable.

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 an efficiently solvable problem on a directed graph.)

34.5-6

Show that the hamiltonian-path problem is NP-complete.