CSC373 Worksheet 7

August 14, 2020

1. **CLRS 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.

2. **CLRS 34.1-3:** Give a formal encoding of directed graphs as binary strings using an adjacencymatrix representation. Do the same using an adjacency-list representation. Argue that the two representations are polynomially related.

3. **CLRS 34.1-4:** Is the dynamic-programming algorithm for the 0-1 knapsack problem that is asked for in Exercise 16.2-2 a polynomial-time algorithm? Explain your answer.

4. **CLRS 34.1-5:** Show that if an algorithm makes at most a constant number of calls to polynomialtime 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.

5. **CLRS 34.1-6:** Show that the class P, viewed as a set of languages, is closed under union, intersection, concatenation, complement, and Kleene star. That is, if $L_1, L_2 \in P$, then $L_1 \cup L_2 \in P$, $L_1 \cap L_2 \in P$, $L_1 L_2 \in P$, $L_1 \in P$ and $L_1^* \in P$.

6. **CLRS 34.2-1:** Consider the language GRAPH-ISOMORPHISM D fhG1; G2i W G1 and G2 are isomorphic graphsg. Prove that GRAPH-ISOMORPHISM 2 NP by describing a polynomial-time algorithm to verify the language.

7. **CLRS 34.2-3:** Show that if HAM-CYCLE 2 P, then the problem of listing the vertices of a hamiltonian cycle, in order, is polynomial-time solvable.

8. **CLRS 34.2-5:** Show that any language in NP can be decided by an algorithm running in time $2^{\mathcal{O}(n^k)}$ for some constant k.

9. **CLRS 34.2-8:** Let be a boolean formula constructed from the boolean input variables $x_1, x_2, , ..., x_k$, negations ($\neq$), ANDs ($\wedge$), ORs ($\vee$), and parentheses. The formula Φ is a tautology if it evaluates to 1 for every assignment of 1 and 0 to the input variables. Define TAUTOLOGY as the language of boolean formulas that are tautologies. Show that TAUTOLOGY $\in$ co-NP.

10. **CLRS 34.2-9:** Prove that P $\subseteq$ co-NP.

11. **CLRS 34.2-10:** Prove that if NP $\neq$ co-NP, then P $\neq$ NP.

12. **CLRS 34.3-3:** Prove that $L \leq_p \bar{L}$ if and only if $\bar{L} \leq_p L$.

13. **CLRS 34.3-7:** Show that, with respect to polynomial-time reductions (see Exercise 34.3-6), L is complete for NP if and only if $\bar{L}$ is complete for co-NP.

14. **CLRS 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. Formulate a related decision problem, and show that the decision problem is NP-complete.

15. **CLRS 34.5-8:** In the **half 3-CNF satisfiability problem**, we are given a 3-CNF formula with n variables and m clauses, where m is even. We wish to determine whether there exists a truth assignment to the variables of 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.