CS091M4041H: Assignment 6

Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

December 19, 2015

Notice:

1. **Due** 9:00 a.m., Jan. 8, 2016 for hard copy and 11:55 p.m., Jan. 8, 2016 for digital version, for graduate students at UCAS;
2. Please submit your answers in hard copy AND submit a digital version to UCAS website https://www2.ucas.ac.cn/.
3. You can arbitrarily choose two from Problems 1-7.
4. In this assignment, only the following problems which have been proved in NP-complete in slides of this course can be used as the well-known NP-complete problem, in order to avoid circular reasoning:
   - SAT
   - 3SAT
   - CLIQUE
   - Independent Set
   - Vertex Cover
   - Set Cover
   - Subset Sum
   - 3 Coloring
   - Hamiltonian Cycle

1 Integer Programming

Given an integer $m \times n$ matrix A and an integer m-vector b, the Integer programming problem asks whether there is an integer n-vector x such that $Ax \geq b$. Prove that Integer-programming is in NP-complete.

2 Mine-sweeper

This problem is inspired by the single-player game Mine-sweeper, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden mine or is empty. The player chooses nodes, one by one. If the player chooses a node containing a mine, the player loses. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (A neighboring node is one connected to the chosen node by an edge.). The player wins if and when all empty nodes have been so chosen.

In the **mine consistency problem**, you are given a graph G, along with numbers labeling some of G's nodes. You must determine whether a placement of mines on the remaining nodes is possible, so that any node v that is labeled m has exactly m neighboring nodes containing mines. Formulate this problem as a language and show that it is NP-complete.

3 Half-3SAT

In the Half-3SAT problem, we are given a 3SAT formula ϕ with n variables and m clauses, where m is even. We wish to determine whether there exists an assignment to the variables of ϕ such that exactly half the clauses evaluate to false and exactly half the clauses evaluate to true. Prove that Half-3SA problem is in NP-complete.

4 Solitaire Game

In the following solitaire game, you are given an $n \times n$ board. On each of its n^2 positions lies either a blue stone, a red stone, or nothing at all. You play by removing stones from the board so that each column contains only stones of a single color and each row contains at least one stone. You win if you achieve this objective.

Winning may or may not be possible, depending upon the initial configuration. You must determine the given initial configuration is a winnable game configuration. Let **SOLITAIRE** $= \{< G > |G$ is a winnable game configuration$\}$. Prove that **SOLITAIRE** is NP-complete.

5 Directed Disjoint Paths Problem

The **Directed Disjoint Paths Problem** is defined as follows. We are given a directed graph G and k pairs of nodes $(s_1, t_1), (s_2, t_2), \cdots, (s_k, t_k)$. The problem is to decide whether there exist node-disjoint paths $P_1, P_2, \cdots, P_k$

so that P_i goes from s_i to t_i. In details, the node-disjoint paths means that P_i and P_j $(1 \leq i \leq k, 1 \leq j \leq k, i \neq j)$share no nodes.

Show that **Directed Disjoint Paths** is NP-complete.

6 Longest Common Subsequence Problem

Given a finite sequence $S = s_1, s_2, \cdots, s_m$, we define a subsequence S' of S to be any sequence which consists of S with between 0 and m terms deleted (e.g. *ac*, *ad*, and *abcd* are all subsequences of *abcd*). Given a set $R = \{S_1, S_2, \cdots, S_p\}$ of sequences, we speak of a *Longest Common Subsequence* of R, LCS(R), as a longest sequence S such that is S is a subsequence of S_i, for $i = 1, \cdots, p$. For example, abe = LCS({ababe, cabe, abdde}).

The *yes/no* **Longest Common Subsequence Problem (LCS)** is: Given an integer k and a listing of the sequences $R = \{S_1, S_2, \cdots, S_p\}$, is $|\text{LCS}(R)| \geqslant k$, where $|S|$ is the number of terms in sequence S?

Show that *yes/no* **LCS** problem is NP-complete.

7 Your favorite intelligence game

Choose one of your favorite intelligence games such as Air Solitaire, Sokoban, Sudoku, 2048 or other games.

Firstly please formulate the game. Maybe the rules of the game is too complex to formulate. To make it easy to analyze, you can simplify the rules or alter the objective of the game while the essential properties of game are still preserved.

Then please analyze the complexity of the game. Give a strategy with polynomial time complexity or prove that the game is NP complete.

You can refer to the Second problem **Mine-Sweep** in this problem set for an example.