ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2020

Problem Set 3

Due: 23:59, Apr. 20, 2020

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name.
3. If you want to submit a handwritten version, scan it clearly.
4. When submitting your homework in Gradescope, match each of your solution to the corresponding problem number.

Note: When proving problem A is NP-complete, please clearly divide your answer into three steps:

(1) Prove that problem A is in NP.

(2) Choose an NP-complete problem B. For any B instance, construct an instance of problem A. Show that the construction runs in polynomial time.

(3) Prove that the yes/no answers to the two instances are the same.

Problem 1:

We learned that 3-SAT is NP-complete in class. Please prove 4-SAT (i.e., each clause contains exactly four literals and the literals in the same clause correspond to different variables) is also NP-complete.

Problem 2:

Suppose you are going to schedule courses for the SIST and try to make the number of conflicts no more than K. You are given 3 sets of inputs: $C = \{...\}, S = \{...\}, R = \{\{...\}, \{...\}, ...\}$. C is the set of distinct courses. S is the set of available time slots for all the courses. R is the set of requests from students, consisting of a number of subsets, each of which specifies the courses a student wants to take. A conflict occurs when two courses are scheduled at the same slot even though a student requests both of them. Prove this schedule problem is NP-complete.
Example:

$$K = 1; \quad C = \{a, b, c, d\}, \quad S = \{1, 2, 3\}, \quad R = \{\{a, b, c\}, \{a, c\}, \{b, c, d\}\}$$

An acceptable schedule is:

$$a \to 1; \quad b \to 2; \quad c, d \to 3;$$

Here only one conflict occurs. An unacceptable schedule is:

$$a \to 1; \quad b, c \to 2; \quad d \to 3;$$

Here two ($> K$) conflicts occur.

Problem 3:

SIST allows students to work as TAs but would like to avoid TA cycles. A TA cycle is a list of TAs $(A_1, A_2, \ldots, A_k)$ such that A_1 works as a TA for A_2 in some course, A_2 works as a TA for A_3 in some course, $\cdots$, and finally A_k works as a TA for A_1 in some course. We say a TA cycle is simple if it does not contain the same TA more than once. Given the TA arrangements of SIST, we want to find out whether there is a simple TA cycle containing at least K TAs. Prove this problem is NP-complete.

Problem 4:

Consider the Knapsack problem. We have n items, each with weight a_j and value c_j $(j = 1, ..., n)$. All a_j and c_j are positive integers. The question is to find a subset of the items with total weight at most b such that the corresponding profit is at least k (b and k are also integers). Show that Knapsack is NP-complete by a reduction from Subset Sum.

Problem 5:

The binary quadratic programming problem can be stated as follows. Given a matrix $A \in \mathbb{Z}^{m \times n}$ and a vector $b \in \mathbb{Z}^m$, is there an $x \in \{0, 1\}^n$ such that $Ax \le b$? (Note: $x \in \{0, 1\}^n$ means x is a vector with n elements and each element is either 0 or 1.)
HINT: Reduction from 3-SAT.

Problem 6:

Given a set E and m subsets of E, $S_1, S_2, ...S_m$, is there a way to select k of the m subsets such that the selected subsets are pairwise disjoint?

Show that this problem is NP complete. HINT: Reduction from Independent Set.