ShanghaiTech University

CS101 Algorithms and Data Structures

Fall 2019

Homework 13

Due date: 23:59, December 26th, 2019

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your FULL Name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
5. When submitting, match your solutions to the according problem numbers correctly.
6. No late submission will be accepted.
7. Violations to any of above may result in zero score.
8. In this homework, all the proofs need three steps. The demand is on the next page. If you do not answer in a standard format, you will not get any point.

Demand of the NP-complete Proof

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 and for any B instance, construct an instance of problem A.
3. Prove that the yes/no answers to the two instances are the same.

0. Proof Example

Suppose you are going to schedule courses for the SIST and try to make the number of conflicts on more than K. You are given 3 sets of inputs: $C = \{\cdots\}, S = \{\cdots\}, R = \{\{\cdots\}, \{\cdots\}, \cdots\}$. 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 species the course 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.

1. Firstly, for any given schedule as a certificate, we can traverse every student's requests and check whether the courses in his/her requests conflicts and count the number of conflicts, and at last check if the total number is fewer than K, which can be done in polynomial time. Thus the given problem is in NP.
2. We choose 3-coloring problem which is a NP-complete problem. For any instance of 3-coloring problem with graph G, we can construct an instance of the given problem: let every node v becomes a course, thus construct C; let every edge (u, v) becomes a student whose requests is $\{u, v\}$, thus construct R; let each color we use becomes a slot, thus construct S; at last let K equals to 0.
3. We now prove G is a yes-instance of 3-coloring problem if and only if (C, S, R, K) is a yes-instance of the given problem:
    - "$\Rightarrow$": if G is a yes-instance of 3-coloring problem, then schedule the courses according to their color. Since for each edge (u, v), u and v will be painted with different color, then for each student, his/her requests will not be scheduled to the same slot, which means the given problem is also a yes-instance.
    - "$\Leftarrow$": if (C, S, R, K) is a yes-instance of the given problem, then painting the nodes in G according to their slots. Since $K = 0$, then for every student, there is no conflict between their requests, which suggests that for every edge (u, v), u and v will not be painted with the same color. It is also a yes-instance of 3-coloring problem.

Therefore, the given problem is NP-complete.

1. (2*1') True or False

(a) Suppose a NP problem is proved to be solved by an algorithm in polynomial time, then it indicates that NP=P.

(a)

(b) For any NP problem, the NPC problem can be reduced to that problem since it is the definition of NPC.

(b)

2. (2*4') NP

Show the following problems are in NP.

(a) Given a graph with n nodes and a number k, are there k nodes that form a clique?(vertices in a clique are all connected to each other)
Part(A): Construct the verifier(1')
Part(B): If the instance x has a solution, show that your verifier works(1')
Part(C): If the instance x has no solution, show that your verifier works(1')
Part(D): Show that verifier works in polytime(1')

(b) Given a set of n cities, and distances between each pair of cities, is there a path vist each city exactly once, and has distance at most D, for a given D?
Part(A): Construct the verifier(1')
Part(B): If the instance x has a solution, show that your verifier works(1')
Part(C): If the instance x has no solution, show that your verifier works(1')
Part(D): Show that verifier works in polytime(1')

3. (★ 10') Knapsack Problem

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. (Subset Sum Problem: Given n natural numbers $w_1, \cdots, w_n$ and an integer W, is there a subset that adds up to exactly W?)

4. (★★ 10') Zero-Weight-Cycle Problem

You are given a directed graph G = (V,E) with weights w_e on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete by a reduction from the Directed-Hamiltonian-Cycle problem.

5. (★★★ 10') Subgraph Isomorphism

Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be isomorphic if there is a one-to-one mapping $f : V \to V'$ such that $(v, w) \in E$ if and only if $(f(v), f(w)) \in E'$. Also, we say that G' is a subgraph of G if $V' \subseteq V$,and $E' = \{(u, v) \in E | u, v \in V'\}$. Given two graphs G and G',show that the problem of determining whether G' is isomorphic to a subgraph of G is NP-complete. (K-clique Problem: Given a graph with n nodes, whether there exist k nodes that are all connected to each other?)