

CS101 Algorithms and Data Structures

Fall 2020

Homework 13

Due date: 23:59, December 21, 2020

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 the above may result in zero grade.
8. In this homework, all the proofs need three steps. The demands and an example are given on the next page. If you do not organize your answer in the 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.

Proof Example

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: $C = \{\dots\}$, $S = \{\dots\}$, $R = \{\{\dots\}, \{\dots\}, \dots\}$. 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 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: (3'+4'+3') Single Choice

The following questions are single choice questions, each question has **only one** correct answer. Select the correct answer.

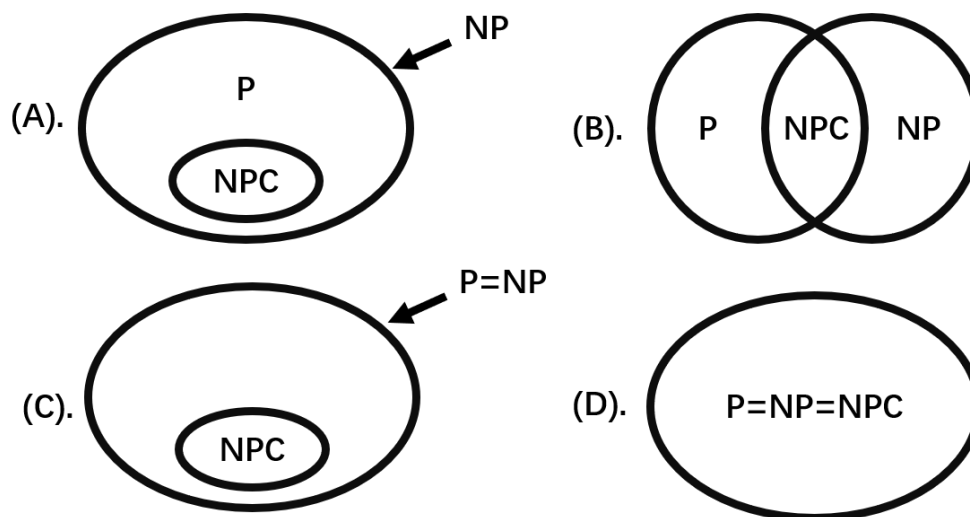
*Note: You should write those answers **in the box** below.*

Question 1	Question 2	Question 3

Question 1. For problems X and Y , Y is NP-complete and X reduces to Y in polynomial time. Which of the following is **true**?

- (A) If X can be solved in polynomial time, then so can Y .
- (B) X is NP-complete.
- (C) X is NP-hard.
- (D) X is in NP, but not necessarily NP-complete.

Question 2. Suppose Prof. Zhao has found an algorithm which correctly solves the vertex cover problem in polynomial time. Then under this circumstance, which one of the following Venn diagrams correctly represents the relationship among the complexity classes P , NP and NP Complete (NPC)?



Question 3. A problem in NP is NP-complete if:

- (A) It can be reduced to the 3-SAT problem in polynomial time.
- (B) The 3-SAT problem can be reduced to it in polynomial time.
- (C) It can be reduced to any other problem in NP in polynomial time.
- (D) Some problem in NP can be reduced to it in polynomial time.

2: (10') 4-COLOR

Given an undirected graph and 4 different colors, can we color the nodes so that no adjacent nodes have the same color? Show that the 4-COLOR problem is NP-complete. (**Hint: Reduce from 3-COLOR**)

3: (10') TA cycle

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, \dots, 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, \dots , 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.

(Hint: Reduce from directed Hamiltonian cycle.)

4: (10') Lecture Planning Problem

You've been asked to organize a freshman-level seminar that will meet once a week during the next semester. The plan is to have the first l weeks consist a guest lecture for each week, and then there are p projects for the students to choose to do.

There are n speakers overall, and in week number i ($i = 1, 2, \dots, l$), a subset L_i of these speakers is available to give a lecture. For each project j ($j = 1, 2, \dots, p$), there is a subset P_j of relevant speakers. Students who chose project j need to see a lecture by **at least one of** the speakers in the set P_j , in order to be able to complete the project.

Given these sets, can you select exactly one speaker for each of the first l weeks of the seminar, so that for each project j , the students will be able to see at least one of the speakers in the relevant set P_j ? Prove this lecture planning problem is NP-complete. (**Hint: Reduce from vertex cover.**)

Example: Let $l = 2, p = 3$, and there are $n = 4$ speakers denoted by A, B, C, D . Availability of the speakers is given by the sets $L_1 = A, B, C$ and $L_2 = A, D$. The relevant speakers for each project are given by the sets $P_1 = B, C, P_2 = A, B, D$, and $P_3 = C, D$. One yes-instance is to choose B for the first week and D for the second week.