ShanghaiTech University

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: $C = \{\cdots\}, S = \{\cdots\}, R = \{\{\cdots\}, \{\cdots\}, \cdots\}$. 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 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
D	D	B

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**?*

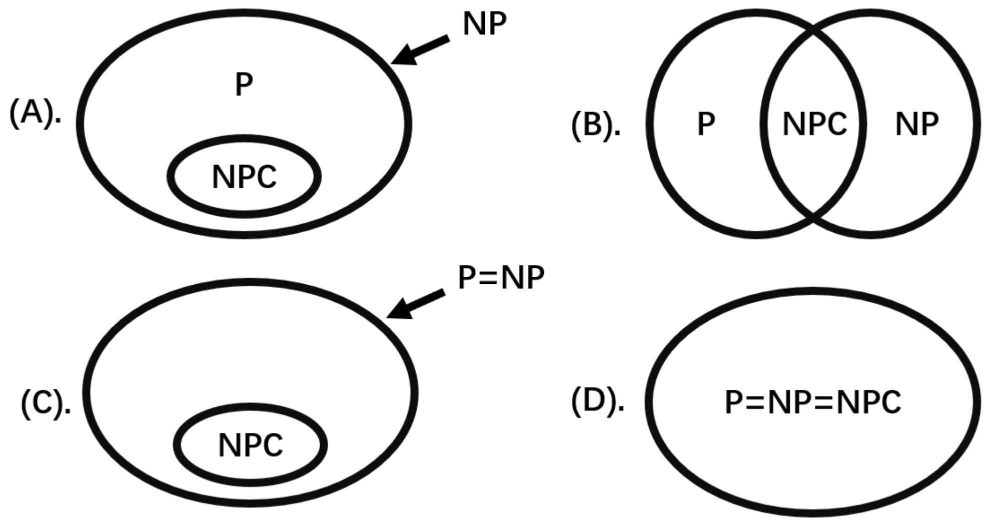
(A) If X can be solved in polynomial time, then so can Y.

(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)**

Solution:

1. First show that 4-COLOR is in NP:

   let *certificate* be a coloring plan. For any given instance, we can check all pairs of two adjacent vertices to see if they have the same color, which has time complexity $O(|V|^2)$ and is in polynomial time. So 4-COLOR is in NP.

2. Choose 3-COLOR to reduce from. For any 3-COLOR instance, we can construct a 4-COLOR instance by adding another vertex that connects all the other points.

3. Now prove a yes-instance of 3-COLOR will lead to a yes-instance of 4-COLOR by such construction, and vice versa:

   - "$\Rightarrow$": For any yes-instance of 3-COLOR, in the 4-COLOR instance, then use the color plan to color the origin vertices, and color the new vertex with a new color, then there is a yes-instance for 4-COLOR.
   - "$\Leftarrow$": For any yes-instance of the constructed 4-COLOR: As the new added vertex is connected to all the other vertices, then it should be a color that not the same with all the other vertices. The other vertices then actually use 3 kinds of colors, which means there is a yes-instance for the 3-COLOR.

   Therefore, the given problem is NP-complete.

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 $A_1, A_2, \cdots, A_k$ 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.
(Hint: Reduce from directed Hamiltonian cycle.)

Solution:

1. First prove that this problem is in NP:

   Given a set of k TAs and all TA relationships as a certifier, we can traverse this TA set and verify whether or not those k TAs form a cycle. This traverse costs $O(k)$ time so it is in NP.

2. Choose directed Hamiltonian cycle problem which is NP-complete. Given an instance of directed Hamiltonian cycle problem $G = (V, E)$: for any edge $(a, b) \in E$, let a be a TA for b.

3. Now prove that there is a simple TA cycle containing at least $|V|$ TAs if and only if G has a directed Hamiltonian cycle:

   - "$\Rightarrow$": If we have a yes-instance of a directed Hamilton circle problem, i.e. there are edges: $v'_1 \Rightarrow v'_2, v'_2 \Rightarrow v'_3, \cdots, v'_k \Rightarrow v'_1$, by the above construction we can see that v'_1 is TA for v'_2 and v'_2 is TA for v'_3, $\cdots$, and finally v'_k is TA for v'_1. This is a TA cycle containing k TAs, so it's also a yes-instance for TA cycle problem.
   - "$\Leftarrow$": If we have a yes instance for a TA-cycle problem, assume there are k TAs involved in this cycle, and v'_1 is TA for v'_2 and v'_2 is TA for v'_3, $\cdots$, and finally v'_k is TA for v'_1. Then by the construction, there are edges: $v'_1 \Rightarrow v'_2, v'_2 \Rightarrow v'_3, \cdots, v'_k \Rightarrow v'_1$, which involves k vertices and forms a cycle. it is a yes-instance of directed Hamilton circle problem.

   Therefore, the given problem is NP-complete.

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 $i(i = 1, 2, \cdots, l)$, a subset L_i of these speakers is available to give a lecture. For each project $j(j = 1, 2, \cdots, p)$, 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.
Solution:

1. First prove that this problem is in NP:

   Given an arrangement $N_1, N_2, \cdots, N_i$, We can first check if $N_1 \in L_1, N_2 \in L_2, \cdots, N_i \in L_i$, and then check if $\{N_1, N_2, \cdots, N_i\} \cap P_1 \neq \emptyset, \{N_1, N_2, \cdots, N_i\} \cap P_2 \neq \emptyset, \cdots, \{N_1, N_2, \cdots, N_i\} \cap P_j \neq \emptyset$. The above steps can all be done in polynomial time, so the lecture planning problem is in NP.

2. Choose vertex cover which is NP-complete. Given an instance of vertex cover $G = (V, E)$ and a number k, we create a lecturer z_v for each node v; let $l = k$, and define $L_1 = L_2 = \cdots = L_k = \{z_v : v \in V\}$ (i.e. for the first $l = k$ weeks, all lectures are available). Then we create a project j for each edge $e_j = (v, w)$, and let $P_j = \{z_v, z_w\}$.

3.
   - "$\Rightarrow$": if there is a vertex cover S of at most k nodes, then consider the set of lecturers $Z_S = \{z_v : v \in S\}$. For each project P_j, at least one of the relevant speakers belongs to Z_S, since S covers all edges in G. Moreover, we can schedule all the lecturers in Z_S during the first k weeks. Thus it follows that there is a feasible solution to the instance of Lecture Planning.
   - "$\Leftarrow$": Suppose there is a feasible solution to the instance of Lecture Planning, and let T be the set of all lecturers who speak in the first k weeks. Let X be the set of nodes in G that correspond to lecturers in T. For each project P_j, at least one of the two relevant speakers appears in T, and hence at least one end of each edge e_j is in the set X. Thus X is a vertex cover with at most k nodes.

   Therefore, the given problem is NP-complete.