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)
F

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

(b)
T

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 certifier(1’)

Denote the certifier as $C(G, S, k)$, where G denotes the original graph, S denotes a certificate (constructed by nodes) and k is the given number. The certifier checks three conditions:

1. Whether S has exactly k nodes
2. Whether S is a subset of G
3. Whether all nodes in S are connected to each other in G

The certifier returns TRUE if all these conditions are satisfied and returns FALSE otherwise.

Part(B): If the instance x has a solution, show that your certifier works(1’)

If there is a solution, then it means there is a clique of k nodes. Let this clique be the certificate S and by definition of clique, all three conditions mentioned above are satisfied for S, so the certifier returns TRUE.

Part(C): If the instance x has no solution, show that your certifier works(1’)

Since there’s no solution, then there’s no clique of k nodes in instance x. For any certificate S there is at least one condition mentioned above that cannot be satisfied, so the certifier always returns FALSE.

Part(D): Show that certifier works in poly-time(1’)

Given a certificate, the certifier needs to check every pair of nodes and every edge between them. Denote the number of nodes as $|N|$ and the number of edges as $|E|$, then the certifier needs $O(|N|^2 \cdot |E|)$. Since $|N|$ and $|S|$ are both polynomial, thus the certifier works in poly-time.

(b) Given a set of n cities, and distances between each pair of cities, is there a path visit each city exactly once, and has distance at most D, for a given D?

Part(A): Construct the certifier(1')

Denote the certifier as $C(G, S, D)$, where G denotes the graph already obtained from the original data (denote each city as a node and connect each pair of nodes with a weighted edge whose value is the distance between corresponding cities), S denotes a certificate (a path) and D is the given distance. The certifier checks three conditions:

1. Whether S has a distance of at most D
2. Whether S is a path in G
3. Whether S is a path that visits each node in G exactly once

The certifier returns TRUE if all these conditions are satisfied and returns FALSE otherwise.

Part(B): If the instance x has a solution, show that your certifier works(1')

If there is a solution, then it means there is a path of at most D distance that visits each city exactly once. Let the digital representation of this path be the certificate S and all three conditions mentioned above are satisfied for S, so the certifier returns TRUE.

Part(C): If the instance x has no solution, show that your certifier works(1')

Since there's no solution, then there's no path of at most D distance that visits each city exactly once. For any certificate S there is at least one condition mentioned above that cannot be satisfied, so the certifier always returns FALSE.

Part(D): Show that certifier works in poly-time(1')

The certifier checks every node and edge in S, since the number of nodes and edges is polynomial, the certifier works in poli-time.

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?)

Solution:

1. Firstly, for any given certificate, the certifier needs to adds up all weights a_j and compare the sum with the weight limit b; the certifier also needs to adds up all values c_j and compare the sum with the profit limit k. These two operations both take $O(n)$, which is poli-time. Thus the given problem is in NP.

2. Choose the Subset Sum problem which is a NP-complete problem itself. For any instance of Subset Sum problem, construct an instance of the given problem: for each integer w_j in $w_1, \ldots, w_n$, let $a_j = c_j = w_j$ and let both weight limit and profit limit $b = k = W$.

3. Now prove an instance (denote as T) of Subset Sum problem is an yes-instance iff the corresponding instance (denote as S) of Knapsack problem is an yes-instance.

   - "$\Rightarrow$": If T is an yes-instance, there is a subset of $\{w_1, \ldots, w_n\}$ that adds up to W, denote that subset as $\{w_{j_1}, \ldots, w_{j_t}\}$ where $1 \le j_1 \le \cdots \le j_t \le n$. Then $\sum_{m=1}^{t} a_{j_m} = W$ and $\sum_{m=1}^{t} c_{j_m} = W$ because of the way we construct S instance, so S is an yes-instance.
   - "$\Leftarrow$": If S is an yes-instance, there is a series of integers $1 \le j_1 \le \cdots \le j_t \le n$ such that $\sum_{m=1}^{t} a_{j_m} \le b = W$ and $\sum_{m=1}^{t} c_{j_m} \ge k = W$. Since $a_{j_m} = c_{j_m} = w_{j_m}$, therefore, $\sum_{m=1}^{t} w_{j_m} = W$ and T is an yes-instance.

Therefore, the given problem is NP-complete.

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.

Solution:

1. Firstly, for any given certificate, we can traverse all cycles in the given certificate and check whether there exists a cycle that the sum of the edge weights is exactly 0. There are polynomial number of cycles so the certifier needs poli-time. Thus the given problem is in NP.
2. We choose the Directed Hamiltonian Cycle problem which is a NP-complete problem. For any instance of Directed Hamiltonian Cycle problem, we construct an instance of the given problem: add an edge from v_i to v_j ($i, j \in [1, n]$) with the weight $1 - n$ and set all the other edges' weights to 1. We do so for n^2 times to find all possible combinations.
3. We now prove an instance (denote as T) of Directed Hamiltonian Cycle problem is an yes-instance iff the corresponding instance (denote as S) of Zero Weight Cycle problem is an yes-instance.
   - "$\Rightarrow$": If T is an yes-instance, then there is a directed hamiltonian cycle (suppose the cycle path is $v_{m_1}, \dots, v_{m_n}, v_{m_1}$) in T and we can find exactly one edge in the corresponding cycle in S that have the weight of $1 - n$ and all the other edges with weights of 1. Since there there will be one combination that an edge of weight $1 - n$ from v_{m_1} to v_{m_n} is added, thus S is an yes-instance.
   - "$\Leftarrow$": If S is an yes-instance, then there is a zero weight cycle (suppose the cycle path is $v_{m_1}, \dots, v_{m_n}, v_{m_1}$) in S. Then in S, there is a directed hamiltonian cycle that has the same cycle path and by changing combinations of added edge, there can be found the corresponding directed hamiltonian cycle in T, thus T is an yes-instance.

Therefore, the given problem is NP-complete.

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?)

Solution:

1. Firstly, for any given certificate, we can traverse all nodes and edges in G and G' to try to make a successful mapping. Since there are poli-number nodes and edges, this can be done in polynomial time. Thus the given problem is in NP.

2. We choose K-clique problem which is a NP-complete problem. For any given instance of K-clique problem with graph G_0, we construct an instance of the given problem: let $G = G_0$ and let G' be a graph with k nodes, where k is the number of nodes in the given instance.

3. We now prove an instance (denote as T) of K-clique problem is an yes-instance iff the corresponding instance (denote as (G, G')) of Subgraph Isomorphism is an yes-instance.

   - "$\Rightarrow$": If T is an yes-instance of K-clique problem, then there is a k-clique in T and also there is a k-clique in G. Since k-clique is isomorphic to any other k-clique, thus the any graph with k nodes is isomorphic to part of the k-clique, so G' is isomorphic to a subgraph of G thus (G, G') is an yes-instance.
   - "$\Leftarrow$": If (G, G') is an yes-instance, then G' is isomorphic to a subgraph of G. Since G' is any graph with k nodes, G' can be the k-clique which means there is a k-clique in G, meaning there is also a k-clique in T. Thus T is an yes-instance.

Therefore, the given problem is NP-complete.