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.

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

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

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

(b)
F

The statement is wrong. For any NP problem, it can be reduced to a NPC problem.

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
Certificate y is a set of k nodes in x. Check each pair of the k nodes is connected by an edge. If so, output 1. Otherwise, output 0.
Part(B): If the instance x has a solution, show that your verifier works
Then there are k nodes that are mutually connected. Call this set y and give it to V. Clearly V outputs 1.
Part(C): If the instance x has no solution, show that your verifier works
Then in any set of k nodes, some 2 nodes aren't connected. So V outputs 0, no matter what set of k nodes it gets.
Part(D): Show that verifier works in polytime
Checking k nodes are mutually connected takes $O(k^2)$ time.

(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
Certificate y is a path through the graph. Check y goes through every city once, and the total length of y is $\leqslant$. If so, output 1, else output 0.
Part(B): If the instance x has a solution, show that your verifier works
Then there is a path going through each vertex once with total length $\leqslant$ D.Call the path y and give it to V. Clearly V outputs 1.
Part(C): If the instance x has no solution, show that your verifier works
Then no matter what path y you use, either y doesn't go through each city once, or y has length $>$ D. So V outputs 0, no matter what y it gets.
Part(D): Show that verifier works in polytime
If the graph has n vertices, then all of V's checks can be done in O(n) 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?)

1. Firstly, given a subset of the items, it is obvious that we can check whether the total weight ia at most b and the corresponding profit is at least k, which can be done in polynomial time. Thus the Knapsack problem is in NP.

2. We choose subset sum problem which is a NP-complete problem. For any instance of subset sum problem with sum W, we can construct an instance of the Knapsack problem with $a_i = c_i = S_i$ and $b = k = W$ which has a satisfying result iff it is satisfiable of subset sum problem.

3. We now prove (W, S) is a yes-instance of subset sum problem if and only if (a, c, b, k) is a yes-instance of the constructed Knapsack problem:

   - "$\Rightarrow$": if (W, S) is a yes-instance of subset sum problem, we can let $\{x_1, x_2, \cdots, x_k\}$ be a result of the problem with $\sum S_i = W$. Then we have $\sum a_i = \sum c_i = b = k$ which means the constructed Knapsack problem is also a yes-instance.
   - "$\Leftarrow$": if (a, c, b, k) is a yes-instance of the constructed Knapsack problem with a result $\{x_1, x_2, \cdots, x_k\}$, then $\sum S_i = \sum a_i \leq b$ and $\sum S_i = \sum c_i \geq k$. Then we have $\sum S_i = W$, which is also a yes-instance of subset sum problem.

Therefore, the Knapsack 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.

1. Firstly, for any given cycle as a certificate, we can check it by going through the cycle to see whether its sum of weights equals to zero, which can be done in polynomial time. Thus the zero-weight-cycle problem is in NP.
2. We choose directed-hamiltonian-cycle problem which is a NP-complete problem. For any instance of DHC problem with graph G, we can produce polynomial number of instances G' of ZWC: for each edge e in G, we let $w_e = -(|V| - 1)$ and others with $w_{e'} = 1$. If any one of these instanced of ZWC return yes, the instance of DHC is yes-instance, otherwise it returns no.
3. We now prove G is a yes-instance if and only if there exists one G' is a yes-instance of ZWC:
   - "$\Rightarrow$": if there is a DHC in G, then when one of the edge is assigned to weight $-(|V| - 1)$ in G', this cycle will have zero weight since the other $|V| - 1$ edges have weight 1.
   - "$\Leftarrow$": if there exists one G' which has a ZWC, it must contain the edge with weight $-(|V| - 1)$ since it is the only one with negative weight and since all the other edges have weight 1, the cycle must contain other $|V| - 1$ edges, which suggests that it will contain all $|V|$ vertices, that will be a DHC in G.

Therefore, the ZWC 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?)

1. Firstly, for any given subgraph as a certificate, we can check whether G' is isomorphic to this subgraph. As graph isomorphism is an NP problem, determining whether G' is isomorphic to a subgraph of G is in NP.
2. We choose CLIQUE problem which is a NP-complete problem. For any instance (G, k) of CLIQUE, where G has n vertices (we set $n \geq k$, otherwise it's an NO instance obviously), we produce the following instance of subgraph isomorphism: (G', G) where G' is a complete graph with k vertices.
3. - "$\Rightarrow$": If G has a clique H with k vertices, then H is a subgraph of G and also a complete graph with k vertices. Considering all complete graphs with k vertices are naturally isomorphic to each other, H is isomorphic to G', thus we can say that G' is isomorphic to a subgraph of G.
   - "$\Leftarrow$": If G' is isomorphic to a subgraph of G, then we denote this subgraph as H. As G' is a complete graph with k vertices, then H is also a complete graph with k vertices (all complete graphs with k vertices are only isomorphic to another complete graph with k vertices). Therefore we can say that G has a yes instance of $k - CLIQUE$ problem.

Therefore, we have shown that subgraph isomorphism is NP-complete.