CSE 431/531: Algorithm Analysis and Design — Fall 2022

Homework 6 Solutions

Instructor: Shi Li — **Deadline: 12/11/2022**

Your Name: ____________________ Your Student ID: ______________

Problems	1	2	3	Total
Max. Score	24	24	32	80
Your Score				

Problem 1 For each of the following 4problems, state (i) whether the problem is known to be in NP, and (ii) whether the problem is known to be in Co-NP. For problems (1b) and (1d), if your answer for (i) (or (ii)) is yes, you need to give the certificate and the certifier that establishes that the problem is in NP (or Co-NP).

(1a) Given a graph $G = (V, E)$ and $s \leq |V|$, the problem asks whether G contains an independent set of size s.

NP: Yes. Co-NP: No.

(1b) Given two circuits C_1 and C_2, each with m input variables $z_1, z_2, \cdots, z_m$, decide if the two circuits compute the same function. That is, whether C_1 and C_2 give the same output for every boolean assignment of z-variables.

NP: No. Co-NP: Yes.

Certifier for the proof for Co-NP: The certifier takes the two circuits C_1 and C_2, and an assignment z of boolean values to the m input variables, and checks if the outputs of C_1 and C_2 are different for the input z.

Certificate for the proof for Co-NP: The certificate is the assignment z for which the outputs of C_1 and C_2 are different.

(1c) Given a graph $G = (V, E)$, decide if G is 3-colorable.

NP: Yes. Co-NP: No.

(1d) Given a graph $G = (V, E)$, decide if G is 2-colorable.

NP: Yes. Co-NP: Yes.

Certifier for the proof for NP: The problem is in P. So the certifier just takes the graph G, and checks if it is 2-colorable in polynomial time.

Certificate for the proof for NP: No certificate is needed.

Certifier for the proof for Co-NP: The problem is in P. So the certifier just takes the graph G, and checks if it is not 2-colorable in polynomial time.

Certificate for the proof for Co-NP: No certificate is needed.

Problem 2 Let NPC be the set of NP-Complete problems. Prove the following statements:

- If P $\neq$ NP, then P $\cap$ NPC $= \emptyset$.

  Assume towards contradiction that $P \neq NP$ and $P \cap NPC \neq \emptyset$. Let $X \in P \cap NPC$. Let Y be any problem in NP. As X is NP-complete, we have $Y \leq_P X$. As $X \in P$, we have $Y \in P$. Therefore every problem $Y \in NP$ is also in P. We have $NP \subseteq P$. But we know $P \subseteq NP$. So $P = NP$. Contradiction.

- If P $=$ NP then P $=$ Co-NP.

  Assume $P = NP$. Let X be any problem in $Co\text{-}NP$. Then $\bar{X} \in NP$ by the relationship between NP and $Co\text{-}NP$. So, $\bar{X} \in P$, i.e., there is a polynomial time algorithm that solves $\bar{X}$. The same algorithm with output negated can be used to solve X. So, we have $X \in P$.

  Therefore $Co\text{-}NP \subseteq P$; but we know $P \subseteq Co\text{-}NP$. Thus $P = Co\text{-}NP$.

Problem 3 In the class, you learned that we can solve shortest path problem on graphs with negative weights, with one caveat: If the graph contains a negative cycle, we can not force the algorithm to find the *shortest simple path* from s to t (a simple path is a path that does not visit a vertex twice).

In this problem, you need to prove that HP $\leq_P$ Shortest-Simple-Path, where Shortest-Simple-Path is the shortest simple path problem, and HP is the Hamiltonian path problem. (Though we did not prove this in class, it is known that HP is NP-complete. So this suggests Shortest-Simple-Path is also NP-complete.)

You can assume the Shortest-Simple-Path problem is a decision problem: we are given a graph $G = (V, E)$ with (possibly negative) edge weights $w : E \to \mathbb{R}$, two different vertices $s, t \in V$ and a threshold $L \in \mathbb{R}$. We need to decide if the shortest simple path from s to t in G has weight at most L or not.

Also, recall that in the HP problem, we are given a graph $G = (V, E)$ and two different vertices s and t, and we need to output if there is a Hamiltonian path between s and t in G or not.

Given an HP instance $(G = (V, E), s, t)$, we define a Shortest-Simple-Path instance (G, s, t, w, L) as follows. We simply let the weight $w(e)$ of every edge $e \in E$ be -1 and let $L = -(n-1)$. ($n = |V|$ is the number of vertices in G.) So, in the instance, we need to know whether there is a simple path in G from s to t of weight at most $-(n-1)$ or not. As all weights are -1, this is the same as whether there is a simple path in G from s to t with at least $n-1$ edges or not. Such a path must be a Hamiltonian path in G from s to t as G contains only n vertices.

Thus, G contains a Hamiltonian path from s to t if and only if the shortest simple path from s to t in G has cost at most $-(n-1)$. This proves HP $\leq_P$ Shortest-Simple-Path.