Study Group – NP-completeness

Name: ____________________ Wisc id: ____________________

The Citadel's Riddle of Complexity

The Citadel of Oldtown, with its towering libraries and arcane knowledge, has long been a beacon of enlightenment in Westeros. Amid its hallowed halls, a cadre of novices has found themselves enthralled by the intricate dance of complexity theory. With the realm's future ever hanging by a delicate thread, these scholarly aspirants seek to harness NP-Completeness, a concept that could unlock new understandings of prophecy and power.

The Archmaesters, venerable sages with intellects as vast as the Summer Sea, have been guiding these novices through the labyrinthine enigmas of computational lore. The novices have crafted a series of queries, fundamental questions on NP-completeness, that probe the depths of their burgeoning wisdom.

1. All problems in NP can be reduced to NP-hard problems in polynomial time. T — F
2. All NP-complete problems can be reduced to each other in polynomial time. T — F
3. If $X \leq_p Y$ and $X \in \mathsf{P}$, then $Y \in \mathsf{P}$. T — F
4. If $X \leq_p Y$ and $Y \leq_p Z$, then $X \leq_p Z$. T — F
5. If 3-SAT $\leq_p X$, then $X \in$ NP-Complete T — F
6. If $X \in \mathsf{P}$, then X can be reduced to 3-D Matching. T — F

   In their scholarly pursuit, one novice has unearthed an ancient scroll that speaks of a deterministic algorithm with the formidable power to resolve any instance of 3-SAT in $O(n^{23} \log n)$ time. The implications of such a discovery are profound, rippling through the very fabric of their academic doctrines.

   In the shadow of this profound revelation, the novice stands before the conclave of Archmaesters, presenting his findings with a mix of trepidation and excitement. He beseeches the wisdom of the ages to discern the veracity of the scroll's bold assertions, framing his inquiry with these pivotal considerations:

7. Any NP problem is solvable in polynomial time. T — F
8. Any NP-Complete problem is solvable in polynomial time. T — F
9. Any NP-Hard problem is solvable in polynomial time. T — F
10. Any NP-Complete problem is solvable in $O(n^{25})$ time. T — F

The Wheel of Westeros: The Zero-Weight Cycle Conundrum

In the intricate web of Westerosi alliances, allegiances shift like sand beneath the feet, where the great houses engage in an eternal dance for power, often described as the turning of a wheel. This wheel can be seen as a vast network of cause and effect, where each action or alliance contributes either positively, negatively, or neutrally to a house's fortunes.

As a master strategist in the service of the Iron Throne, you have been presented with a map that intricately details the current state of alliances, betrayals, and neutral pacts among the great houses. Each pact has been assigned a weight that represents its impact: positive for beneficial alliances, negative for costly betrayals, and zero for neutral relations. There might be more than one possible choice of pacts between houses i and j.

The problem task is to find a sequence of pacts that begins and ends with the same house and whose combined impact is neutral, symbolizing the 'wheel' of power that perpetually turns but ultimately leads back to where it began. This is known as the **Zero-Weight Cycle problem**. Prove that it is NP-Complete

Parameters:

- G: Multi-graph.
- W_{ijk}: The weight of pact k between house i and house j, which can be positive, negative, or zero.

Hint: Consider a reduction from Subset Sum

Intractability recap

Polynomial-Time Reductions

$Y \leq_p X$

- Consider any instance of problem Y.
- Assume we have a black-box solver for problem X.
- Efficiently transform an instance of problem Y into a polynomial number of instances of X that we solve via black-box solver for problem X, and aggregate the solutions efficiently to solve Y.

Corollaries from polynomial time reductions:

- Suppose $Y \leq_p X$. If X is solvable in polynomial time, then Y can be solved in polynomial time.
- Suppose $Y \leq_p X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.
- Suppose $Z \leq_p Y$ and $Y \leq_p X$. Then, $Z \leq_p X$.

Intractability Definitions

Decision Problems

- Binary output: yes / no answer.
- Our complexity definitions assume decision problem versions of the problems.
- No less powerful: we can go between decision and optimization version of problems.

Easy vs Hard Problems

- *Easy Problems*: problems that can be solved by efficient algorithms.
- *Hard Problems*: problems for which we do not know how to solve efficiently.

Input Formalization

- Let s be a binary string that encodes the input.
- $|s|$ is the length of s, i.e., the # of bits in s.

Complexity class: P

- Polynomial run-time: Algorithm A has a *polynomial run-time* if run-time is $O(\mathsf{poly}(|s|))$ in the worst-case, where $\mathsf{poly}(\cdot)$ is a polynomial function.
- P is the set of all problems for which there exists an algorithm A that solves the problem with polynomial run-time.

Efficient Certification: Certifier $B(s,t)$ for a problem P:

- s is an input instance of P.
- t is a certificate; a proof that s is a yes-instance.
- Efficient: For every s, we have $s \in P$ iff there exists a t, $|t| \leq \mathsf{poly}(|s|)$, for which $B(s,t)$ returns yes.

Complexity class: NP

- Set of all problems for which there exists an efficient certifier $B(s,t)$.
- I.e., the set of all problems for which it is efficient to verify a potential solution.
- **N**on-deterministic, **P**olynomial time: can be solved in polynomial time by testing every certificate (t) simultaneously (non-deterministic).

Complexity class: NP**-Hard:** Problem X is NP-Hard if:

- For all $Y \in \mathsf{NP}$, $Y \leq_p X$.
- NP-Hard problem may or may not be in NP.

Complexity class: NP**-Complete:** Problem X is NP-Complete if:

- For all $Y \in \mathsf{NP}$, $Y \leq_p X$.
- X is in NP.

Showing that Problem X is NP-Complete: Cook Karp Reduction

Step 1: Prove that $X \in \mathsf{NP}$.

(a) Define a certificate (t) for X.

(b) Define an efficient certifier (algorithm) $B(s,t)$ for X and t as defined in (a).

Step 2: Choose a problem $Y \in \mathsf{NP}$-Complete.

- Y must be a problem that is known to be NP-Complete.
- It will be used to show that $Y \leq_p X$ in step 3:
  - Since $Y \in \mathsf{NP}$-Complete, then all NP problems $\leq_p Y$.
  - Therefore, showing $Y \leq_p X \implies$ all NP problems $\leq_p X \implies X \in \mathsf{NP}$-hard.

Step 3: $Y \leq_p X$ ($X \in \mathsf{NP}$-hard).

- Karp Reduction: For an arbitrary instance s_Y of Y, show how to construct, in polynomial time, an instance s_X of X such that s_y is a yes iff s_x is a yes.
  Steps:
  (a) Provide efficient reduction.
  (b) Prove $\Rightarrow$: if s_Y is a yes, s_X is a yes.
  (c) Prove $\Leftarrow$: if s_X is a yes, then s_Y had to have been a yes.