NP-Complete Chapter Knowledge

**Polynomial Algorithms:** on inputs of size n, their worst-case running time is O() for some constant k.

## -- Difference between optimization problem input/output to algo and decision problem input/output to algo

**Decision problems versus optimization problems**

**Decision Problems:** It asks whether there exists a feasible solution that satisfies a certain condition or problem. The answer is yes or no.

**Input:** Does a problem instance satisfy some certain conditions?

**Output:** Yes or no

**Optimization problems:** An optimization problem involves finding the best solution from all feasible solutions.

**Input:** What is best value k of the problem instance with some certain conditions?

**Output:** The output is the best value k, like minimal or maximal value

## -- P, NP, NP-C, NP-H

**Class P:** P is a set of decision problems that are polynomial-time solvable on a deterministic sequential machine.

**Class NP:** NP is a set of decision problems whose solutions can be verified in polynomial time, and whose solution can be found in polynomial time on a non-deterministic machine.

**Class NP-C:** A decision problem L is NP-Complete if

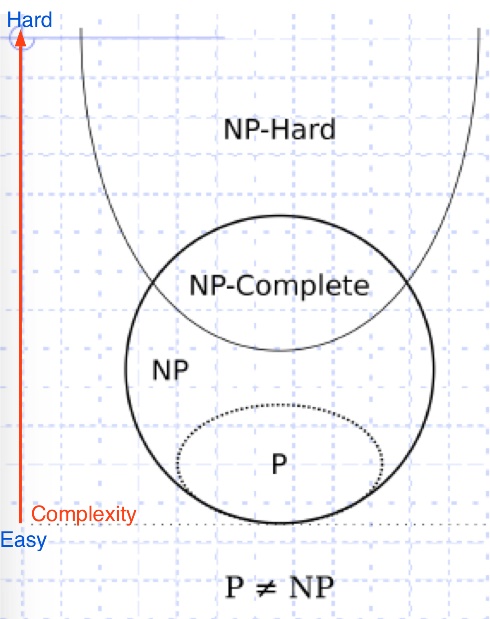
1: L ∈ NP, and

2: for every L’ ∈ NP, L’ L

**Class NP-H:** If a decision problem L only satisfies property 2, but not necessarily property 1, we say that L is NP-Hard.

## -- Is P = NP or P a subset of NP? explain the importance of this question and common believe and why, draw picture of this and explain your picture

Is P = NP or P a subset of NP? explain the importance of this question and common believe and why,



Most of people believe P ≠ NP, but P ⊆ NP. This belief is based on the fact that many attempts to find polynomial-time algorithms for NP-Complete problems have been unsuccessful so far.

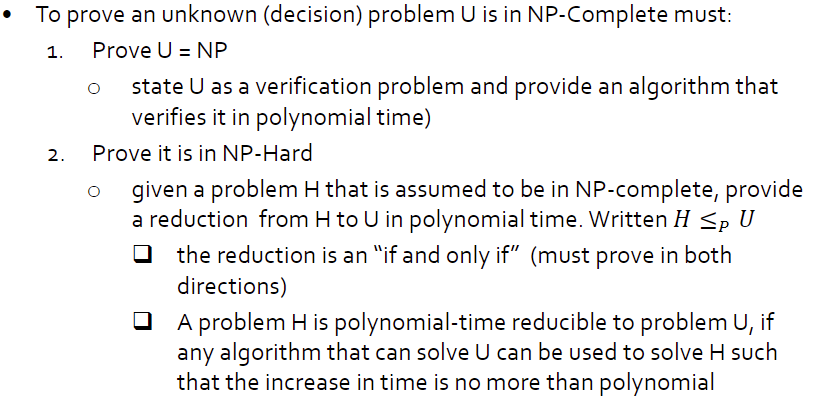
If P = NP, it means that every problem in NP can be solved by a deterministic Turing machine in polynomial time.

If P is a proper subset of NP, it means there are problems in NP for which no polynomial-time algorithm exists.

## -- What are the steps to prove a problem is in NP-Complete?

1: Prove it is NP

2: Prove it is NP-Hard



-- Be able to prove a decision problem is in NP-Complete. Focus on the ones we did in class.

-- Be able to give problem definitions.

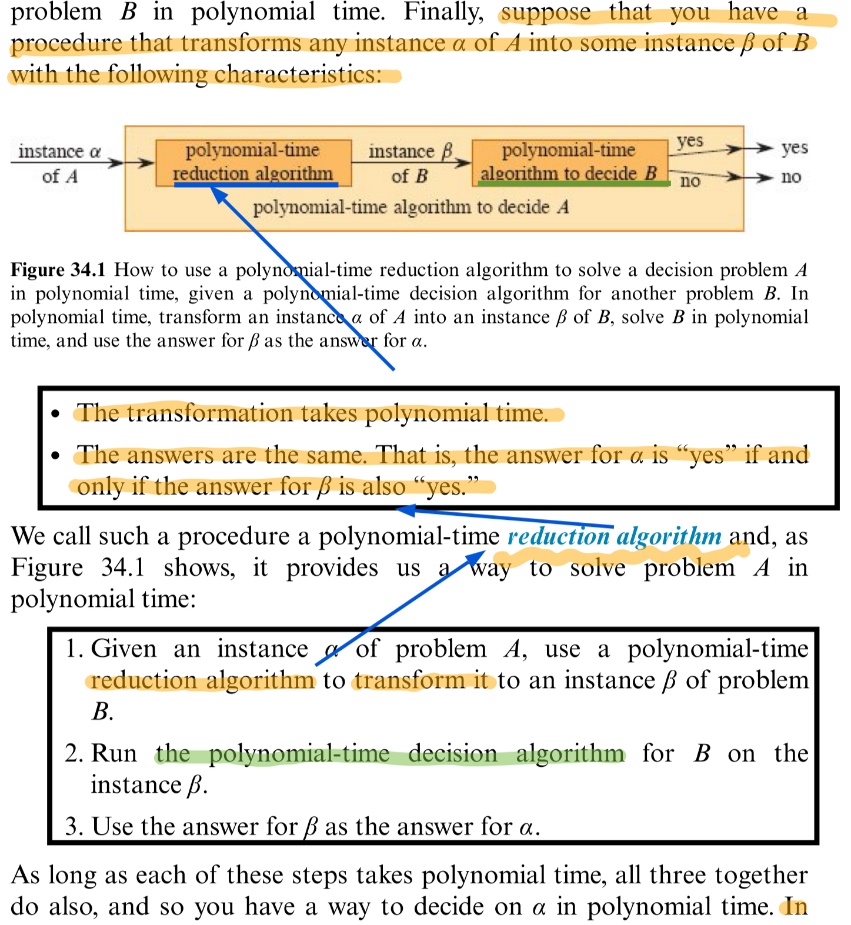
-- Be able to pose a problem as a decision problem and a verification problem and an optimization problem, when appropriate.

## -- Explain the difference between a decision and verification problem statement

**Decision Problems:** It asks whether there exists a feasible solution that satisfies a certain condition or problem. The answer is yes or no.

**Verification problem:** It is concerned with **verifying** whether **a given solution** to a problem is correct or valid.

**Reduction**

****

