

CSC373 Worksheet 7 Solution

August 14, 2020

1. Notes

- **Decision Problem**

- Is the problem if determining answer to a class of yes/no questions about some objects of interest

- **Reduction Problem**

–

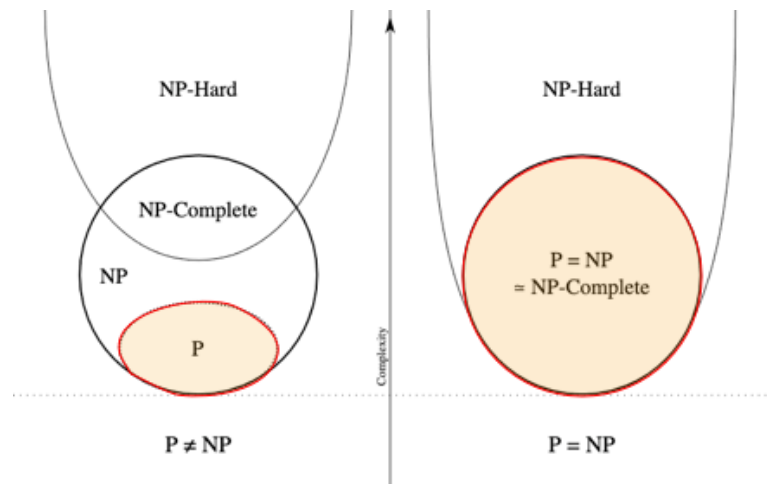
Example:

- **P**

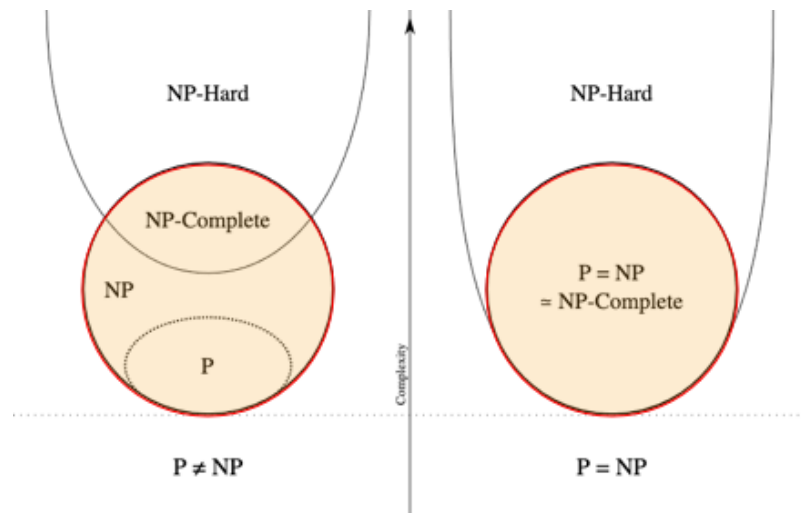
- Is set of problems that can be solved by a deterministic Turing machine in Polynomial time (i.e. $\mathcal{O}(n^k)$) [2].

Example:

- 1) Shortest path problems
- 2) Calculating the greatest common divisor
- 3) Finding maximum bipartite matching



- **NP (Non-deterministic Polynomial):**



- Is set of decision problems that can be solved by a Non-deterministic Turing Machine in Polynomial time.^[2]
- Has no particular rule is followed to make a guess ^[1].
- Can be solved in polynomial time via a “lucky algorithm”, a magical algorithm that always make a right guess ^[2]
- $P \subseteq NP$

Examples:

- Longest-path problems
- Hamiltonian Cycle
- Graph coloring

- **NP-Complete Problems:**

- A decision problem A is NP-complete (NPC) if
 - 1) $A \in NP$ and
 - 2) Every (other) problems A' in NP is reducible to A
- Has no efficient solution in polynomial number of steps (not yet) ^[3]
- Is not likely that there is an algorithm to make it efficient ^[3]

- **NP-Hard:**

- A decision problem A is NP-hard if
 - 1) $A \in NP$ (Not necessarily) and
 - 2) Every (other) problems A' in NP is reducible to A
- NP-Hard means “at least as hard as any problems in NP”
- Does not have to be about decision problems

Example:

- 1) Alan Turing's Halting Problem

References

- 1) Encyclopedia Britannica, NP-Complete Problem, [link](#)
- 2) Geeks for Geeks, NP-Completeness, [link](#)
- 3) Wikipedia, NP-complete, [link](#)
- 4) UCLA UC-Davis, ECS122A Handout on NP-Completeness, [link](#)