Complexity Theory

Vimal Raj Sharma

Lecture 1
Course Overview

Overview of complexity theory:

- Qn: What is computation?
  Ans: Producing output from input in a finite number of steps.
- MULT: Given two numbers x and y, compute $x \cdot y$.
  - Design a fast algorithm to compute $x \cdot y$. (Algorithms)
  - Prove that no faster algorithms exist. (Complexity Theory)
- Resources for computation: Time, Space, Interactions, Random Bits, etc.
- Central goal of Complexity Theory: Proving non-existence of efficient algorithms for problems.
- What we actually do in Complexity Theory:
  - Prove non-existence of efficient algorithms. (E.g., `GeneralisedChess` $\notin$ P.)
  - Interrelate different complexity questions.
    - Are L_1 and L_2 not solvable in polytime?
    - L_1 is not polytime solvable $\iff$ L_2 is not polytime solvable.
  - Classify problems based on the amount of resources required to solve them and compare those classes.

    X = Set of problems solvable in logspace.

    Y = Set of problems solvable in polytime.

    Z = Set of problems solvable in polyspace.

    $$X \subseteq Y \subseteq Z$$
    $$(X \neq Y\,?,\ Y \neq Z\,?,\ X \neq Z)$$

- Computability Theory vs Complexity Theory

(deals with proving non-existence of any algorithms) (deals with proving non-existence of efficient algorithms)

Glimpses of this course:

- Is P = NP?

  P = Set of problems that are polytime solvable.

  NP = Set of problems whose solutions are polytime verifiable.

  Examples:

  PATH: Given a graph G and $u, v \in G$, find whether $u \rightsquigarrow v$.

  HAMPATH: Given a graph G, find whether a path exists that consists of all the vertices of G.

- Are there problems solvable in $O(n^3)$ time that are not solvable in $O(n)$ time?

- Problems beyond NP. Example:

  INDSET: Given a graph G and an integer k, find whether G has an independent set of size k. ($\in$ NP)

  EXACT-INDSET: Given a graph G and an integer k, find whether the size of the largest independent set of G is k. ($\in \Sigma_2^p$)

- Given a directed graph G and $u, v \in G$, can we find whether $u \rightsquigarrow v$ in logspace? (Is L = NL?)

- Can we use randomness to speed up the computation?

  P = Set of problems that are polytime solvable by deterministic algorithm.

  BPP = Set of problems that are polytime solvable by probabilistic algorithm.

  Examples:

  1. PRIMES: Is x prime? (Is in BPP. Not known to be in P in past, but in P presently (AKS'02))
  2. PIT(Polynomial Identity Testing): Given a multivariate polynomial with integer coefficients find whether there is an assignment of values to variables such that polynomial evaluates to non-zero. (Is in BPP, but not known to be in P.)

*detailed syllabus on mail.

Grading:

- 5% - Class Participation.
- 15% - 20 minutes presentation on a paper/topic in groups of two or single.
- 20% - Best two out of three quizzes. (Mostly MCQs and T/F).
- 30% - Minors (15% each).
- 30% - Major.

Books:

- Computational Complexity: A Modern Approach by Arora and Barak.
- Computational Complexity: A Conceptual Perspective by Goldreich.
- Introduction to the Theory of Computation by Sipser.

Office Hours:

Wednesday: 4-6 PM. (From next week.)

Course Site:

https://home.iitj.ac.in/~vimalraj/courses/csl7140

Attendance:

As per institute policy.