Introduction to Computational Complexity

Introduction This course is a graduate level elective that introduces topics in computational complexity. Computational complexity is a study of the resources necessary and sufficient to solve a given set of problems. It is a very active area of research. We will study many classical results in the first part of the course. In the second part we will study advanced topics in complexity theory.

Pre-requisites We assume basic knowledge in linear algebra, combinatorics, probability, algorithms design and analysis, and automata theory (DFA/NFA etc.).

Credit Structure Crediting/Auditing the course:
The following credit structure is likely to change slightly. However, by the end of the 3rd week of instruction, when the course registrations are stabilized, it will be finalized.

Credit

problem sets/quizes	20%	(approx)
mid-sem	25%	
end-sem	35%	
seminar (group activity)	20%	(approx)

Audit

4 problem sets	65%	(approx) 2 before mid-sem and 2 after mid-sem
end-sem	35%	

Course Outline In the first part of the course, we will study the following topics.

1. Introduction to the course, Turing Machines, Time hierarchy theorem.
2. P, NP, NP hardness, Cook-Levin theorem.
3. Space complexity classes - L, NL, PSPACE, NPSPACE, Savitch's theorem, Immerman-Szelepcsényi – proof of NL=coNL.
4. Turing machines with randomization, BPP, examples of problems in BPP, non-uniform circuits and BPP $\subseteq$ P/poly.

In the second part of the course, we will explore one of the following research themes.

1. Algebraic complexity theory and lower bounds.

2. Boolean circuit complexity and lower bounds.
3. Communication complexity and information theory.
4. PCP theorem and applications.

Seminar Topics Depending on the class strength (the number of students crediting the course), we will group the students and give them topics to read and present. Every group will be expected to give a clear presentation (possibly using electronic presentation), and will be expected to generate an electronic report (preferably using LaTeX).

References

- Introduction to the Theory of Computation – Michael Sipser
- Computational Complexity: A Modern Approach – Sanjeev Arora and Boaz Barak

  `www.cs.princeton.edu/theory/complexity/book.pdf`
- Introduction to Complexity Theory (Lecture Notes for a Two-Semester course [1999]) – Oded Goldreich

  `http://www.wisdom.weizmann.ac.il/~oded/cc99.html`
- Computational Complexity (Lecture Notes [Fall 2002]) – Luca Trevisan

  `http://www.cs.berkeley.edu/~luca/notes/complexitynotes02.pdf`