Introduction to Computational Complexity

Introduction This course is a graduate level elective that introduces the basic topics in the subject of 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 classic theorems for most part of the course. Towards the end, we will also present some recent results.

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

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

Credit

assignments	15%	(approx) 2 before mid-sem and 2 after mid-sem
scribing notes	10%	(approx) 1 or 2 lectures
mid-sem	25%	
end-sem	35%	
seminar	15%	group activity

Audit

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

Course Outline

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.
5. Boolean circuits and formulas, Brent's depth reduction of formulas, NC, AC, SAC.

6. Structural results for Boolean circuits.
7. Arithmetic circuits, VP, VNP.
8. Structural results for arithmetic circuits.
9. Boolean circuit and formula lower bounds.
10. Arithmetic circuit and formula lower bounds.

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`