RANDOMIZED METHODS IN COMPLEXITY

PROF. NITIN SAXENA
Department of Computer Science and Engineering
IIT Kanpur

PRE-REQUISITES : Preferable (but not necessary)- Theory of Computation, or Algorithms, or Discrete Mathematics

INTENDED AUDIENCE : Computer Science & Engineering, Mathematics, Electronics, Physics, & similar disciplines

INDUSTRIES APPLICABLE TO : Discrete Optimization, Cryptography, Coding theory, Computer Algebra, Symbolic Computing Software, Cyber Security, Learning Software

COURSE OUTLINE :

In this course we will study how randomness helps in designing algorithms and how randomness can be removed from algorithms. We will start by formalizing computation in terms of algorithms and circuits. We will see an example of randomized algorithms-- identity testing --and prove that eliminating randomness would require proving hardness results. We prove hardness results for the problems of parity and clique using randomized methods.

ABOUT INSTRUCTOR :

He completed his Bachelors in Computer Science from the Indian Institute of Technology, Kanpur in 2002 and completed his PhD under Manindra Agrawal in 2006. He is broadly interested in Computational Complexity Theory, Algebra, Geometry and Number Theory. He has been a visiting graduate student in Princeton University (2003-2004) and National University of Singapore (2004-2005); a postdoc at CWI, Amsterdam (2006-2008) and a Bonn Junior Fellow (W2 Professor) at Hausdorff Center for Mathematics, Bonn (2008-2013). Since April 2013, He has a faculty position in the department of CSE, IIT Kanpur.

COURSE PLAN :

Week 1**:** Outline. Introduction to Complexity

Week 2**:** Circuits. Polynomial Identity Testing (PIT)

Week 3**:** Derandomize & get a lower bound

Week 4**:** Constant-depth circuits are weak

Week 5**:** Monotone circuits are weak

Week 6**:** Random Walk converges fast

Week 7**:** Expansion properties

Week 8**:** Construct Explicit Expanders

Week 9**:** Pseudorandom generator (prg) & hardness

Week 10**:** Error-correcting codes

Week 11**:** List Decoding. Local List Decoding

Week 12**:** Error-correcting codes amplify hardness