USING RANDOMNESS IN COMPUTER SCIENCE

NICOLAS SCHABANEL

1. Introduction to randomized algorithms (definition, cost,...) **[Lec.1]**
2. Simpler (and faster) than deterministic: A randomized algorithm for Min-CUT
   1. A simple $O(n \log n)$-time randomized algorithm that succeeds with probability $\Theta(1/n^2)$
   2. A recursive $O(n^2)$-time randomized algorithm that succeeds with probability $\Theta(1/\log n)$.
   3. Increasing the probability of success to $1 - 1/n^c$ for any constant $c > 0$.
3. Less ressources than deterministic: A $\Theta(\log \log n)$-bits randomized counter that counts up to n
4. Proving optimality of randomized algorithms: Yao's principle **[Lec.2]**
   1. Faster than deterministic: Randomized evaluation of OR-AND complete boolean trees
   2. Yao's principle
   3. A lower bound on randomized algorithm for AND-OR trees based on Yao's principle
   4. Hard drive energy consumption optimization
      1. Optimal deterministic algorithm
      2. Optimal randomized algorithm via Yao's principle **[Lec.3]**
5. Randomized rounding
   1. Approximation algorithms for Max-SAT
      1. The "random solution" randomized algorithm
      2. A randomized algorithm based on linear programming
      3. Combining both
      4. Derandomizing these algorithms by the conditional expectation method
   2. Approximation algorithms for Max-CUT
      1. The "random solution" algorithm
      2. Derandomizing this algorithm by the conditional expectation method yields an analysis of the simple greedy algorithm
6. Sample and Guess **[Lec.4]**
   1. A polynomial-time randomized algorithm for testing polynomial identity (Zero-P)
   2. Self-correction of integer multiplication
   3. The class PCP(r,q); NP $\subseteq$ PCP$(\log n, 1)$ $\Leftrightarrow$ $(\exists c)$ GAP-SAT$_{1,c}$ is NP-hard (i.e. hardness of approximation result for Max-SAT)
   4. Linearity testing and self-correction
   5. (Application) One of the main step of the PCP theorem: NP $\subseteq$ PCP(poly$(n), 1)$ **[Lec.5]**
      1. QUADEQ is NP-complete
      2. A PCP$(n^2, 1)$-verifier for QUADEQ
   6. A constant-time approximation scheme for the size of a maximal matching in a constant degree graph **[Lec.6]**
7. Exhaustive guessing 1
   1. A polynomial-time randomized $(1-\varepsilon)$-approximation for Max-CUT in dense graph
8. Random Walks 1
   1. Walk-SAT

Date: Dec 2010-Feb 2011.