{ Csc 80030 | Lecture 1 }

Probabilistic Analysis & Randomized Algorithms

Hoeteck Wee · hoeteck@cs.qc.edu

Today

- Overview of this course
- Course administration
- A randomized algorithm

Randomization in Computer Science

- Algorithm Design
  - Basic algorithmic problems, e.g. PRIMALITY (1977, 2002)
  - Practical problems, e.g. symmetry breaking in sharing resources
- Cryptography
  - Randomness provides secrecy, e.g. 4-digit PIN random in $\{0000, \ldots, 9999\}$.
- Computational Models
  - Random processes, e.g. natural selection & mutation in biology
  - Complex networks, e.g. social networks and the Internet

This Course

- Randomized algorithms
  - Simplicity: Randomized min-cut, median-finding and 2-SAT
  - Efficiency: Sublinear-time algorithms
  - Average-Case "Goodness": Load balancing
- Tools and techniques for probabilistic analysis
  - Tail bounds, e.g. Markov's inequality and Chernoff bounds
- Computational models
  - Random graphs

Administration

- Basic Information
  - Course webpage `www.cs.qc.edu/~hoeteck/s09`
  - Contacting me `hoeteck@cs.qc.edu`
  - Webpage + email for disseminating information
  - Textbook: Probability and Computing: ..., by Mitzenmacher & Upfal
- Pre-requisites
  - Strong background in basic probability; basic algorithms course
- Course "rescheduling"? 4.10 PM - 6 PM

Course Evaluation

- Homework: $\sim$ once every two weeks
- One mid-term: Mar 18 (maybe 2-4 pm?)
- One programming assignment
- Final project
- Class attendance and participation

Polynomial Identity Testing

Identity Testing

Given two polynomials $p(x)$ and $q(x)$, decide whether $p \equiv q$ (that is, whether p is "identical" to q).

- "polynomials": coefficients are integers or field elements; degree $\leq d$
- "$p \equiv q$": coefficients for each monomial are the same, e.g. $(x+1)(x-1) \equiv x^2 - 1$
- "given": (1) list of coefficients, or (2) as a formula, e.g. $((x-1)^2+1)^3 + 4x$.

Polynomial Identity Testing

Identity Testing

Given two polynomials $p(x)$ and $q(x)$, decide whether $p \equiv q$ (that is, whether p is "identical" to q).

Identity Testing (special case)

Given a polynomial $p(x)$, decide whether $p \equiv 0$.

- To solve the general case, check whether $p(x) - q(x)$ is identical to 0.

Polynomial Identity Testing

Identity Testing Algorithm

1. Pick a number r uniformly at random from $\{1, 2, \ldots, 2d\}$.
2. Evaluate $p(r)$. If the result is 0, accept; else, reject.

- If $p(x) \equiv 0$, then algorithm always accepts.
- If $p(x) \not\equiv 0$, then algorithm accepts with probability $\leq \frac{1}{2}$.

Fact

A non-zero degree d polynomial has at most d roots.

Polynomial Identity Testing

Identity Testing Algorithm

1. Pick a number r uniformly at random from $\{1, 2, \ldots, 2d\}$.
2. Evaluate $p(r)$. If the result is 0, accept; else, reject.

- If $p(x) \equiv 0$, then algorithm always accepts.
- If $p(x) \not\equiv 0$, then algorithm accepts with probability $\leq \frac{1}{2}$.

Question

How can we reduce the error (i.e. the probability of accepting $p(x) \not\equiv 0$)?

Polynomial Identity Testing

1. Try all r in $\{1, 2, \ldots, d+1\}$.
   - Always outputs correct answer.
   - Problem: d may be as large as 2^n. e.g.

$$(((\overbrace{(x+1)^2+1)^2+1)^2\cdots+1}^{\longleftarrow\ n\ \text{times}\ \longrightarrow})^2$$

2. Replace $2d$ with $1000d$.
   - Reduces error to $1/1000$.
   - Disadvantage: need to compute with large numbers.
3. Repeat k times, using different random values r
   - Reduces error to $1/2^k$.
   - Advantage: works in general for any randomized algorithm.

(Almost) the End

- Next week: review basic probability
- Homework 1 to be posted by Fri, due Feb 11 (Wed).
- Short quiz