

Current Topics in Theoretical CS

Lecture 1

Logistics

Instructor: Elena Grigorescu, elena-g@purdue.edu

Course website: <http://www.cs.purdue.edu/homes/egrigore/Fall12/>

Introductions: name, year, background

Assignments:

- scribe notes for 1-2 lectures
- project:
 - full lecture where you present a theory paper
 - + notes for it
 - you can work in pairs
 - meet with me twice before the presentation

Course topics

- Sublinear models of computation
- Error-correcting codes
- Expander graphs
- Communication complexity

Sublinear models of computation

- **Program checking**
Quickly determine whether a piece of code computes the right function
- **Property testing**
Quickly decide whether a mathematical object has a property or is very far from having it
- **Approximation algorithms**
Quickly output an approximate value for the solution of a hard problem
- **Streaming algorithms**
Quickly compute a function of a data stream arriving in an online fashion
- **Local correction/local decoding**
Quickly recover a possibly corrupted value in a string of bits

Quickly = using time/space resources \ll linear in the input size

How? Must use randomness!

Sublinear models of computation

- Have lots of practical applications
- Analyzing these applications require beautiful mathematics

Probabilistic methods

Extremal combinatorics

Fourier analysis

Markov chains

High-dimensional geometry

Sublinear algorithms in this course

- Testing properties of graphs:
 - Testing if a graph contains triangles
Powerful tool: Szemerédi regularity lemma
Further application to additive combinatorics
 - Testing connectedness
Related application: approximating the number of connected components
- Testing properties of functions:
 - Testing monotonicity
Combinatorial tool: spanner graphs
- Local algorithms for error-correcting codes
 - Computational tasks: testing, correcting, decoding
Tools: polynomials
- Streaming algorithm (example in this lecture)

Error-correcting codes

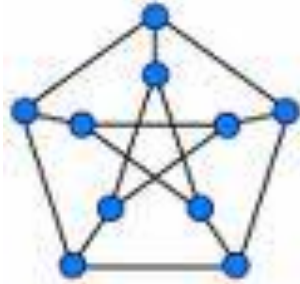
- ECC: way to encode data so that it can be recovered after corruption

0 1 1 \Rightarrow 0 0 0 1 1 1 1 1 1

- Used everywhere in practice:
 - Communication: deep space transmission, cell lines, TV broadcast
 - Storage: CDs, DVDs, clouds
 - Networks protocols, data indexing (bar codes, ISBN), etc.
- Questions:
 - What features can be achieved?
 - Explicit constructions?
 - Algorithms?
 - Tools for applications in computational complexity?
(eg. PCP theorem)
- Techniques: analysis of polynomials over finite fields, expander graphs, etc.

Expander graphs

- Def: Sparse graphs (i.e. with few edges) that are highly connected.



- Applications:
 - Coding theory: building codes with good parameters
 - Study of pseudorandomness in computation
 - Computational complexity (eg. PCP theorem)
 - Cryptography
 - Data structures
 - Distributed computation

Communication Complexity

- Two parties want to compute a function of their input by exchanging only a few bits



- Provides successful techniques for proving lower bounds in
 - Data structures
 - Streaming algorithms
 - Property testing

Other topics?

Your own choice goes here

Tentative schedule

- Introduction to sublinear models of computation
- Probability basics
- Testing connectivity. Estimating the number of connected components and minimum spanning tree weight.
- Szemerédi regularity lemma. Testing triangle freeness in dense graphs. Applications to additive combinatorics.
- Proof of the triangle removal lemma.
- Testing monotonicity of functions. Spanner graphs.
- Intro to codes. Finite fields. Hadamard, Reed-Solomon, Reed Muller.
- Locally testable codes. Testing membership in the Hadamard code (aka linearity testing). The Fourier analytic method.
- Locally decodable codes.
- Expander graphs. Codes from graphs. Other applications.
- Communication complexity. Application to Property Testing lower bounds

The remainder of this lecture

(on the board)

- An example of a sublinear **deterministic** algorithm
 - estimating the diameter of a set of points in a metric space
- A simple example of a streaming algorithm
 - taking a uniform sample from an online stream
- Property testing intro
- Survey