CS5234
Combinatorial and Graph Algorithms

Welcome!

CS5234 Overview

- **Webpage:**
  http://www.comp.nus.edu.sg/~gilbert/CS5234

- **Instructor: Seth Gilbert**
  Office: COM2-323
  Office hours: by appointment

Why are we here?

Why are we here?

Combinatorial and *BIG* Graph Algorithms:

Why are we here?

Combinatorial and *BIG* Graph Algorithms:

All new module… no relation to CS5234 2014.

Motivating question

What happens when we have a graph containing 4 billion nodes and 1.2 trillion edges?

(The facebook graph, for example, is at least that big.)

New Challenges

Scale

- How do we deal with graphs that are very big?
- Cannot store entire graph in memory.
- Processing time is large!

New Challenges

Where is the data?

- Data is no longer as easily accessible.
- Is data distributed?
- Is data streaming?
- Is data noisy?

New Challenges

Dynamic world

- Data is no longer static.
- Graphs change over time.
- Edges may be added and removed.
- Users may come and go.

New Challenges

Context matters

- Where did the data come from?
- Is it from a social network?
- Is it from a wireless network?
- Is it from a game?
- How can we leverage the structure to do better?

Example: Minimum Spanning Tree

Algorithms 101

- Kruskal’s Algorithm
- Prim’s Algorithm
- Runs in $O(m \log n)$ time for n nodes and m edges.
- Fast enough?

Example: Minimum Spanning Tree

Special Structure

- Is graph planar? Then we can find an MST in O(m) time.
- Is the graph a social network?

Example: Minimum Spanning Tree

Randomization and Approximation

- Can we find a faster randomized algorithm?
- What if we only need an approximate minimum spanning tree?
- What if we only want to estimate the size of the MST?
  - Amazingly: $O(dw \log(dw))$ time for a graph with degree d and max. edge weight w.

Example: Minimum Spanning Tree

Streaming

- What if we only have limited access to data?
- We get to read each edge once in some arbitrary order: $e_1, e_2, e_3, \ldots, e_m$
- (Also good for cache.)
- We can’t store the whole graph!
- Then we need to output an (approximate) MST.

Example: Minimum Spanning Tree

Dynamic

- What if edges change over time?
- Edges are continually added and removed from our graph.
- After each change, we need to find a new MST.

Example: Minimum Spanning Tree

Caching

- At these scales, caching performance is critical.
- Each time we access part of the graph, a block of memory is loaded. (That's expensive.)
- How can we design an algorithm for finding an MST that uses cache efficiently?

Example: Minimum Spanning Tree

Parallel/GPU/Distributed

- Can we leverage a multicore machine to find an MST faster?
- Can we use GPUs to get faster performance?
- Can we use a distributed cluster (e.g., MapReduce/Hadoop) to find an MST faster?

Goal

Explore a set of tools for answering these questions.

“If you need your software to run <u>twice as fast</u>, hire better programmers.

But if you need your software to run <u>more than twice as fast</u>, use a better **algorithm.**”

-- *Software Lead at Microsoft*

Goal

Explore a set of useful tools for answering these questions.

See a bunch of neat algorithms.

“... pleasure has probably been the main goal all along.

”

-- *D. E. Knuth*

Comabinatorial and

(BIG) Graph Algorithms

“... pleasure has probably been the main goal all along.

But I hesitate to admit it, because computer scientists want to maintain their image as hard-working individuals who deserve high salaries... ”

-- *D. E. Knuth*

CS5234 : Combinatorial and Graph Algorithms

Brand new class:

- We can make this class what we want.
- Talk to me about your goals, interests, etc.

CS5234 : Combinatorial and Graph Algorithms

Target students:

- Advanced (3rd or 4th year) undergraduates
- Master's students
- PhD students
- Interested in algorithms
- Interested in tools for solving hard problems

Prerequisites:

- CS3230 (Analysis of Algorithms)
- Mathematical fundamentals

CS5234 Overview

- Mid-term exam

  October 6 In class

- Final exam

  November 8

CS5234 Overview

- Lecture

  Thursday 6:30-8:30pm

- Tutorial

  Thursday 8:30-9:30pm

  Tutorial will be used for discussion, reviewing problem sets, answering questions, solving riddles, doing crossword puzzles, eating cookies, etc.

CS5234 Overview

- Grading

  40% Problem sets

  25% Mid-term exam

  35% Final exam

- Problem sets
  - 5-6 sets (about every 1-2 weeks)
  - Focused on algorithm design and analysis.
  - Perhaps a few will require coding.

CS5234 Overview

- Mini-Project

  Small project

  Idea: put together some of the different ideas we have used in the class.

  Time scale: last 4 weeks of the semester.

CS5234 Overview

- Released tomorrow

Survey: On IVLE.

What is your background?

Not more than 10 minutes.

PS1: Released tomorrow.

CS5234 Overview

- Problem set grading

  Simple scheme:

  3 : excellent, perfect answer

  2 : satisfactory, mostly right

  1 : many mistakes / poorly written

  0 : mostly wrong / not handed in

  -1 : utter nonsense

CS5234 Overview

- What to submit:

  Concise and precise answers:

  Solutions should be rigorous, containing all necessary detail, but no more.

  Algorithm descriptions consist of:

  1. Summary of results/claims.
  2. Description of algorithm in English.
  3. Pseudocode, if helpful.
  4. Worked example of algorithm.
  5. Diagram / picture.
  6. Proof of correctness and performance analysis.

CS5234 Overview

- How to draw pictures?

  By hand:

  Either submit hardcopy, or scan, or take a picture with your phone!

  Or use a tablet / iPad…

  Digitally:

  1. xfig (ugh)
  2. OmniGraffle (mac)
  3. Powerpoint (hmmm)
  4. ???

CS5234 Overview

- Policy on plagiarism:

  Do your work yourself:

  Your submission should be *unique*, unlike anything else submitted, on the web, etc.

  Discuss with other students:

  1. Discuss general approach and techniques.
  2. Do not take notes.
  3. Spend 30 minutes on facebook (or equiv.).
  4. Write up solution on your own.
  5. List all collaborators.

  Do not search for solutions on the web:

  Use web to learn techniques and to review material from class.

CS5234 Overview

- Policy on plagiarism:

  Penalized severely:

  First offense: minimum of one letter grade lost on final grade for class (or referral to SoC disciplinary committee).

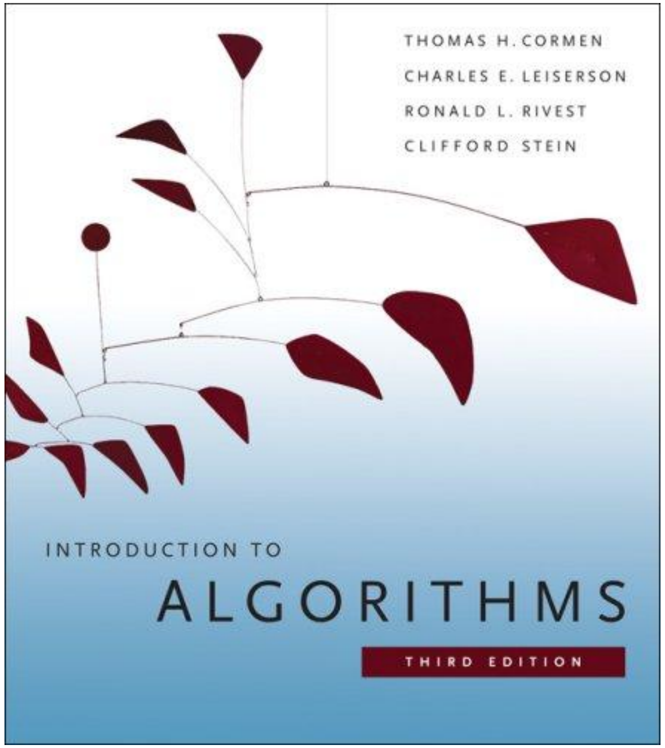
  Second offense: F for the class and/or referral to SoC.

  Do not copy/compare solutions!

Textbooks

Introduction to Algorithms

- Cormen, Leiserson, Rivest, Stein

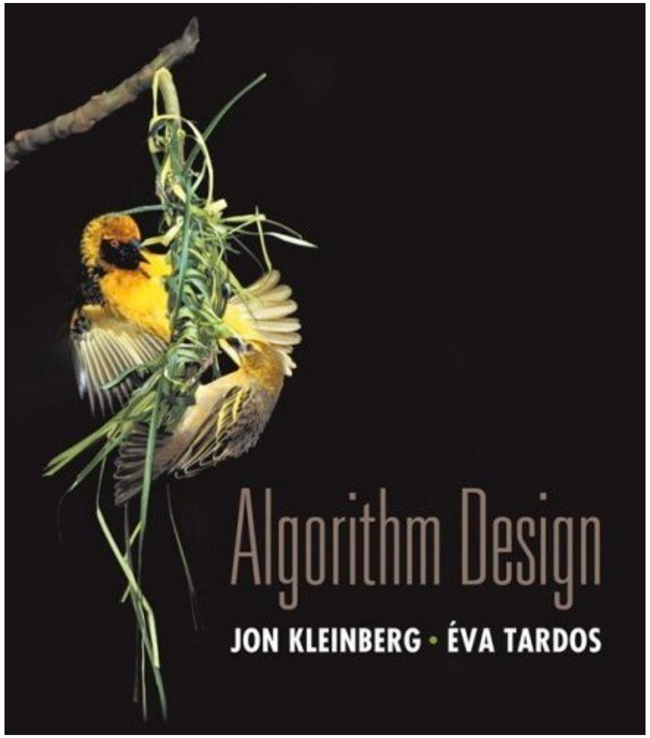


- Recommended…

Textbooks

Algorithm Design

- Kleinberg and Tardos



- Recommended…

CS5234 Overview

- Topics (tentative, TBD)
  - Graph connectivity and Minimum Spanning Trees
    - Sampling from a stream
    - Sketching a graph
    - Sublinear time algorithms
  - Cuts and sparsification
    - Randomized minimum cut
    - Sparsifying a graph
    - Spanners and other useful graph structure

CS5234 Overview

- Topics (tentative, TBD)

  Caches

  - Cache-efficient algorithms
  - Cache-oblivious algorithms
  - How to tune cache performance

  Parallel and distributed algorithms

  - Multicore efficient algorithms
  - Analyzing parallelism
  - MapReduce and other distributed models

CS5234
Combinatorial and Graph Algorithms

Welcome!