Announcements:

10/1: updated office hours are as follows:
Kevin: Monday 10am-12noon (5th floor lounge of CSB).
Pedro: Tue 10:50am-12:50pm (CS TA room).
Alex: Tuesday 2:30-4:30pm.

9/10: TA office hours for this week are as follows.
Pedro: Tue 10:50am-12:50pm (CS TA room),
Kevin: Fri 10am-noon (5th floor lounge of CSB).


If you are on the waitlist and want to be in the class, please send an email to the instructor.

You can use [this cheatsheet](http://piazza.com/columbia/fall2015/comse69989/home), which has a few useful statements, including the Chernoff/Hoeffding bounds.

Administrivia:

Instructor: Alex Andoni. Email: andoni cs.columbia.edu. Office hours: 517CSB Tue 2:30-4:30.
TAs: Pedro Savarese (Tue 10:50am-12:50pm), Kevin Shi (Mon 10am-12noon).
Time: TR 1:10-2:25pm.
Location: 602 Hamilton.

Prerequisites: #4231 (Analysis of Algorithms 1) or equivalent (with the permission of instructor).
The class is based on theoretical ideas and is proof-heavy, hence mathematical maturity is a must. Background in algorithms and some probability theory will be assumed.
If you are an undergraduate student, feel free to contact me regarding getting into the class.


Description:

You may have heard about the promise (and the challenge) of the Big Data, but how exactly are we to extract that promise? The classic algorithmic approaches to processing data are insufficient to deal with the ever-increasing data sizes. For example, a quadratic-time algorithm means that 10x increase in data size requires a 100x increase in resources!

This class will focus on modern algorithmic techniques to tackle massive datasets efficiently. We will see formalized models for some massive data settings, and explore core algorithmic ideas arising in them. Many such ideas are based on the notion of capturing complex data, or partial computation, with smaller but nonetheless useful sketches.

Tentative list of topics:
- streaming: algorithms that see data as a single stream and use space smaller than the data itself to compute its properties. Some particular problems include distinct elements, approximate counting, heavy hitters, frequency moments.

- sketching: algorithms for representing objects approximately with a small "sketch", and how to use such sketches to speed-up algorithms. We will see the concept of dimension reduction, among others.

- high-dimensional nearest neighbor search: preprocess a dataset of objects (e.g., images), so that, given a new query object, one can quickly return similar objects from the dataset. We will see the concept of Locality-Sensitive Hashing, among others.

- sampling (property testing): how to learn something about an object (e.g., distribution) just by sampling it in a few locations.

- parallel algorithms: algorithms in modern parallel models (such as MapReduce) where the computation is accomplished by many computational units that each have limited space.

You can see (preliminary) schedule here.

**Resources and study material:**

See [Lectures and Scribes](#).