NYU Tandon School of Engineering
Computer Science and Engineering
CS 6913, Fall 2019

September 9, 2019

CS 6913: Web Search Engines

Course Prerequisites: Solid Computer Science background. Excellent programming skills, preferably in C/C++. General experience with the web and with HTML is expected. Knowledge of algorithms (equivalent to CS6033 or CS3414) is strongly recommended. Useful but not required are basic knowledge of networking and of Unix network and systems programming, scripting languages, and basic OS and database concepts.

Time and Location: Mondays from 3:20–5:50 pm in room 9.007 in 2 MetroTech.

Instructor: Prof. Torsten Suel, email: `torsten.suel@nyu.edu`, office: room 856 in 370 Jay Street.

Office Hours: T 2-3pm in my office, or by appointment (send email).

Grader: A graduate assistant will grade the homeworks and projects. The assistant will also have office hours and be available via email and Piazza. Details to be announced.

Course Webpage: An up-to-date page will be available at `http://engineering.nyu.edu/~suel/cs6913/` within the next few days. We will also use NYU Classes for important announcements and homework submissions, and Piazza for course discussions.

Textbooks: The following book is recommended as background reading for this course. See also the course page for additional resources.
– Introduction to Information Retrieval, by Manning, Raghavan, and Schutze, Cambridge University Press, 2007 (available online for free).

Grading Policy: The final grade will depend on course assignments (30%), course project (30%), and final exam (40%). The project also requires writing a longer paper describing the work and results.

Final Exam Date: The final exam will be held on Monday, December 16, at 3:20pm. Room TBD.

Policy on Academic Dishonesty: Please read NYU Tandon's policy on academic dishonesty at https://engineering.nyu.edu/campus-and-community/student-life/office-student-affairs/policies/student-code-conduct. Common examples of misconduct include cheating, fabrication, plagiarism, and/or unauthorized collaboration. Students are expected to work on their own, with the possible exception of group projects if allowed by the Professor. Students may discuss work with other individuals either in the class or outside the class, but they may not reuse code, results, or text received or retrieved from any source unless clearly disclosed in their submissions. Any student who is found to be violating this policy will be given a failing grade for the work and will be reported to the authorities, including the CSE Department's student records, as described in the University's Student Code.

Moses Center Statement of Disability: If you are a student with a disability who is requesting accommodations, contact New York University's Moses Center for Students with Disabilities (CSD) at 212-998-4980 or `mosescsd@nyu.edu`. You must be registered with CSD to receive accommodations. See `http://www.nyu.edu/students/communities-and-groups/students-with-disabilities.html` for information about the Moses Center. The Moses Center is located at 726 Broadway on the 2nd and 3rd floors.

Excused Absences: See the NYU Tandon policies and procedures on excused absences located at https://engineering.nyu.edu/campus-and-community/student-life/office-student-affairs/policies#chapter-id-30199. In short, an absence can be excused if you have missed no more than 10 days of school. If an illness or special circumstance has caused you to miss more than two weeks of school, please refer to the section labeled *Medical Leave of Absence*. Students may request special accommodations for an absence to be excused in the following cases: (1) Medical reasons, (2) death in the immediate family,

(3) personal qualified emergencies (documentation must be provided), and (4) religious expression or practice. Deanna Rayment, deanna.rayment@nyu.edu, is the Coordinator of Student Advocacy, Compliance, and Student Affairs, and handles excused absences. She is located in 5 MTC, LC240C, and can assist you should it become necessary.

Assignments: There will be several programming and written assignments. General discussions between students, and help in *setting up and using any programming environments*, are permitted and encouraged. However, no code or solutions may be copied! In the second half, students will work on a larger project, individually or in small groups. A list of available projects will be presented, and students may also propose relevant projects of their own interest.

Course Outline: This course will cover a variety of topics related to web search technology. The main focus will be on large-scale web search engines (such as Google, Bing, or Baidu) and the underlying architectures and techniques. You will learn how search engines work, and get hands-on experience in building search engines from the ground up. You will also learn about the fundamental challenges and bottlenecks in current search technologies, and about research efforts that address these issues. In this context, we will also cover fundamentals of information retrieval, data compression, and computing with massive data sets. Here are some typical topics:

- basic web architecture
- search engine architecture
- web crawling, web exploration, and web surveillance
- indexing of very large text data sets
- text and data compression
- I/O-efficient computing and computing with massive data sets
- search engine query execution and ranking functions
- use of link structure in web search and data analysis
- computational advertising
- search engine manipulation and spam
- query log mining and personalization
- ranking, retrieval models, and learning to rank
- advanced search engine architecture

The course will cover both algorithmic techniques and implementation aspects. Students will be required to complete several substantial programming projects in C/C++, Java, or the Python scripting language. Students will also have to read a number of research papers.

Policies and Suggestions. Following are a few tips on how to approach the assignments and projects.

(1) Start on the assignments early! Assignments may be more substantial than they appear on first sight. But others may be easier, once you figure out the necessary parts. But they all need some planning, so a last-minute approach is discouraged.

(2) Do not reinvent the wheel! For example, Python has a lot of useful functions in its standard modules. Check it out. You many also find other useful libraries on the web. Of course, you should not use a libary that completely solves the entire assignment on its own, say a crawler in the first assignment – if in doubt, ask for permission.

(3) Do not skip programming assignments! You should do all of them. Subsequent assignments may reuse code from earlier ones. Make sure you write your code such that you can modify it easily later.

(4) Discuss the assignments with classmates. Seek and provide help about things such as the programming environment, library functions, design decisions. Don't get stuck for hours because you cannot figure out, e.g., how to run a Python program or how to use a particular library - ask somebody. Ask other students, then ask the instructor if necessary.

(5) But do not cross the line. Do not copy code and solutions from others, and disclose any outside resources you have used as part of the readme file for the assignment.

(6) If you are in serious danger of falling behind, talk to the instructor BEFORE it is too late.

(7) Make use of the office hours. If needed, there will also be special project discussion hours, to meet as a group and discuss any common problems with the assignments.