CS 761: Randomized Algorithms Fall 2019

Final Project

Prof. Gautam Kamath *Deadline: One Week before grades are due (TBD)*

The goal of this project is to apply your new comfort, intuition, and skills with probabilistic reasoning in an algorithmic context, in order to read and possibly undertake your own original research involving randomness and algorithms. There are two options for projects.

- **Reading project.** Find a set of papers on one topic, and write a coherent summary article that synthesizes the ideas. The papers should be fairly modern and somewhat challenging – barring exceptional cases, there should not be a currently written pleasant exposition of the set of results being covered (that's your job). You will be graded based on how much better your exposition and synthesis is in comparison to the original papers.
- **Theory project.** Develop a new and "better" solution to some algorithmic problem, somehow involving randomness and probability. In this context, "better" is a somewhat subjective term: it could mean more efficient by some metric, but it could also mean simpler, or demonstrating a new connection that was not previously known. Any improvement on previous results will be acceptable, but you will be graded on the strength of the improvement.

Note that one might start by undertaking a theory project, but be unable to prove novel results. The project could then be converted to a reading project, on the same topic. A "hybrid" project is also acceptable, especially in such situations. Perhaps you managed to prove something novel, but it's fairly minor – this could be presented within the context of a larger reading project.

Regardless, the deliverable for this project is a roughly 10 page paper describing the results. You will be graded based on presentation and readability. It should be written at a level understandable by anyone in this class: in particular, no background should be assumed besides an undergraduate degree in computer science and the topics covered in this class. Anything else should be covered in the exposition of your paper. If you're not sure whether you can assume something, feel free to ask.

Projects can be done in groups of up to three people. You should submit only one report for your entire group. A "divide and conquer" approach will not work: the point is to synthesize ideas, so every person should have read and understand every paper that goes into the report.

There will be a brief project proposal, due November 8 at 11:59 PM by email to me and the TA. This will involve roughly a paragraph or two, telling me what you plan to study, and who you will be working with. The purpose of this is to just make sure you're not going down a direction that is out of the scope of the class – you are not married to the idea, and you can change it drastically in the future (but ping me to let me know that you're doing so).

Good writing is difficult, good technical writing is even harder. You don't have to provide all details, but the reader should come away with a good picture on what the big ideas are. Think of the level of detail I provide during lectures: while I sometimes enumerate every calculation, I often

The inspiration for this style of project is from the inimitable David Karger, as well as some segments of the text.

leave out some details so we don't get bogged down. Note that I expect more details for a theory project rather than a reading project: it's a bad sign if I can't verify the correctness of your result. If you are concerned about a long technical proof taking away from the presentation, feel free to defer such things to an appendix (though I might not read this very carefully). If you are doing a reading project, one paper is generally not enough, but perhaps two or three are. You need to synthesize the ideas to relate the papers to each other. You should *not* just enumerate the results without discussion of the techniques: you should try to extract the key insights, ideas, and proofs that are needed to arrive at them. In short, the reader should feel like they learned something technical from reading your document. Similarly, just describing algorithms is not that useful – you should provide insight into *why* they work.

Naturally, you should aim for a topic which is somehow related to this class – not necessarily in terms of the specific topics, but at least in the ideas, analysis, and style of thinking which we have developed. Since most of you are graduate students working on research, you might find some interesting algorithmic questions related to your work. That said, you should preferrably not work on something that is *exactly* your current research.

There are many sources of papers which would be relevant to this type of investigation. Some of the big ones are:

- The ACM Symposium on Theory of Computing (STOC)
- The IEEE Symposium on Foundations of Computer Science (FOCS)
- The ACM-SIAM Symposium on Discrete Algorithms (SODA)
- The International Colloquium on Automata, Languages and Programming (ICALP)
- The International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)
- The International Conference on Randomization and Computation (RANDOM)

You are welcome to draw on papers which are outside of these conferences (some of my other favorites include COLT, ICML, and NeurIPS), but with the warning that the further you stray from the list above, the less likely the papers are likely to be within the scope of this class. If unsure, just ask.