

## Instructions for Final Project

*Prof. Arnab Bhattacharyya*

1. Form a group of 1-3 students. You are encouraged to form groups as group projects are usually more fun and you will be able to get more done.
2. Choose either a reading project or an implementation project. In either case, your group will be responsible for a writeup and an interview.
3. For a reading project, your goal is to synthesize information from several papers in an area and develop **your own understanding** of the algorithms and their analyses. The emphasis in the last sentence is there because you will be graded on the novelty in your descriptions of the results. In an ideal project, you will have found your own proof of a particular lemma or your own way to present an algorithm that is different from what's there in the original papers. Your goal should be to convince your fellow classmate of why it all works (even if you can't give all proofs in full detail). You also need to identify at least one open question in your chosen area and give your thoughts about it.

Some tips:

- Look up older papers that are cited as well as more recent papers that cite the papers you are looking at. Google Scholar is your friend!
  - For the open question, try to spend at least 10 days on it. Come up with a conjecture or proposed algorithm, do some simulations if possible, make some partial progress.
  - Understand that older papers may be complicated and later papers may be simpler and more insightful. Don't be intimidated by technical proofs. Try to figure out the high-level ideas driving the calculations and only then attempt to understand the details. Have a notebook and pen handy; keep writing down what you observe.
4. For an implementation project, you will be graded on how well you explain the algorithm you are implementing, what kind of interesting heuristics you developed, what interesting test cases you ran them against, and how well you interpret the results.

Algorithms papers tend to leave out a lot of little implementation details that turn out not to be so little when the time comes to implement. You may also explore implementation of heuristics that have “no theoretical value” but may lead to enormous improvements in practice. Once you have an implemented algorithm, you can test it to see how it performs in practice. This involves devising interesting “hard” inputs that make the algorithm perform poorly. Study of the algorithm's behavior may lead you to make changes in the algorithm and test them. A typical implementation paper says “we implemented algorithm X and it was awful, then we added heuristic Y and it was great!”

Give serious thought to test inputs. Ideally, they will come from real-world problems, or will be specially designed to “stress” some aspect of the algorithm. You also need a control (baseline) algorithm to compare yours to.

5. Form your group and decide what you want to work on. Let me know by email by **Friday, March 13**. Topics are first come, first serve. So it's to your benefit to get started early.
6. Schedule a meeting with me in the week **March 16–21** to discuss details for your specific project.
7. Schedule a final interview with me in the week **April 27 – May 1**. Submit the final report at the start of the interview.