

Final Project Guidelines

You may opt to complete a final project instead of the take-home final. Below are guidelines (intentionally similar to previous years).

- **Your project must involve algorithm design/analysis.** Your project can involve implementation, but there should be a novel algorithmic component.
- You should spend comparable time on your project as you would preparing for a final (e.g. a few weeks).
- You are encouraged to work in groups of two. We discourage larger or smaller groups - you must ask permission to work in a larger group or alone. Piazza is a good place to look for a project partner.
- Submit a 1-page proposal by **Dec 5** (email to cmusco@cs.princeton.edu and kothari@cs.princeton.edu). You will get feedback and we'll discuss if necessary. Feel free to set up a meeting or chat with the instructors anytime before 11/30 for guidance on finding a topic.
- Then, submit a 2-page final proposal by **December 13**.
- Aim for ≈ 10 pages with your final report (but there is no ceiling or floor). You should clearly explain the problem you aimed to solve, any necessary context to appreciate the problem's relevance, and what you accomplished. If you found barriers to accomplishing what you originally desired, describe them clearly as well.
- The final project report will be due on **January 15th** (dean's date).
- We'll have a meeting between **3pm – 5pm** on **January 15** where each group will give a 10-minute presentation on their project.

How to choose a topic:

- Your project should constitute **original work**, but it is not meant to be publishable work. That is, you should do something novel and in the same spirit as a research paper, but you're not expected to produce publishable work in a few weeks.
- You're encouraged to pick a topic close to your own interests (e.g. another area of CS, or something else entirely). But the meat of your project should be algorithmic.
- Appropriate example project 1: Pick an area of interest with publicly available data. Use algorithms we saw in class to analyze the data and learn something interesting. Note that this often requires much more than simply implementing an algorithm - you'll likely need to tweak the algorithms in some way, evaluate their performance, and interpret the results.

- Appropriate example project 2: Pick a topic covered in class and generalize algorithms we saw to handle richer settings (the setting should be somewhat well-motivated).
- In general, try to identify a clean, simple objective that you can reasonably make progress on in a few-week period. Clearly identify what your novel contribution will be, and why this contribution is “algorithmic.”

Try to keep the project simple and exploratory. A primary goal for theory is to simplify and focus on the essentials. (Doesn't always work but very powerful when it does.)