Final Project/Exam Guidelines

December 3, 2021

You may complete a final project or take a final exam. Roughly, you should think of the final exam as (moderately) less time-consuming, but (moderately) higher variance. Below are rough guidelines for both.

1 Final Exam

The final exam will be posted on the course website by **December 3**. You may choose any 48 hour window to complete the exam.

- To begin the exam, send Matt and Mark (smweinberg@princeton.edu, mbraverm@cs.princeton.edu) an email with subject line [COS 521 FA 21 Final] stating that you are starting the exam.
- Within 48 hours (by email timestamp) send us a reply on that same email thread containing your completed final exam.
- The exam absolutely must be submitted by 11:59pm on the university's last day to submit take-home final exercises (**December 20**), independently of when you start.
- You aren't expected to spend (anywhere close to) the full 48 hours on the exam.
- The exam will cover the entire course. Some questions will be "easy", and others will be "challenging." You do not need to get 100% to do "well." In particular, there will be seven questions on the exam, but we will only score the highest six.

2 Final Project

Quick Logistics. To complete a final project, you should have a group of size two or three (larger than three is not permitted, solo projects are strongly discouraged).

- Email Matt and Mark (smweinberg@princeton.edu, mbraverm@cs.princeton.edu) with a brief-but-concrete plan by **November 30**.
- The final project report will be due on dean's date (**December 14**).

- We'll have one zoom meeting to present projects to the class. Fill out this doodle poll to indicate availability for presentation dates by **November 30**. Each group will give a 10-minute zoom presentation on their project.[1]

How to choose a topic. There are (roughly) three kinds of projects that can be good for this course. See this document for some concrete suggestions in each category.

- Try out a project in theoretical algorithm design. For this kind of project, we should pick a topic together (e.g. you should rely more heavily on the suggested topics, and expect to discuss with me/Mark). This is a good choice if you are hoping to do a JP/IW/thesis/research in theoretical algorithm design.

- Complete a project closer to your own research interests, which uses tools from the course. A sample theme might be "find a dataset which is important to your own research. Use algorithms developed in class (which are not already widespread in your field) to dimension-reduce/learn/etc. the dataset. Compare it to existing methods." (You can replace "research interests" with "personal interests" as well, or "analyze a dataset" with any other algorithmic problem resolved by tools in this course). For this kind of project, you should propose a direction, and have your own plan for how to do 'the work'. You should still check in with us to get feedback on your plan.

- Write a good/thorough survey on a narrow field related to this course. If you want to do a theoretical topic, you should rely more heavily on the suggested topics, and expect to discuss with me/Mark. This is also a good choice if you are hoping to do a JP/IW/thesis/research in theoretical algorithm design. If you want to do a topic closer to your own interest (e.g. an application of advanced algorithms in your research area), you should propose your own topic (and again check in with us to get feedback).

For any project, you are **strongly encouraged** to tie the project to your own interests. Ideally, the project itself should be narrow so that you can complete concrete tasks in a short period. If this course intersects your research interests, it is good for the project to have potential to expand into a JP/IW/thesis/submission, in case you really enjoy it (of course, it is also completely fine if you intend to just complete the project and move on). These are bins that we hope most projects will fall into, but you are also welcome to propose other ideas.

Scope of the project. You have ≈ 2 weeks to complete the project, so we are not expecting you to solve a major open problem, or produce a publication. We are, however, expecting you to do *something concrete*. A good guideline to have in mind is that the *execution should be good*, but the *scope can be (very) narrow*. For example, if you pick a theoretical project, this might mean that you prove a very (very) special case of a conjecture, or prove that a very specific proof approach cannot work. If you pick an applied project, this might mean that you produce working code, which you can run on actual input and be evaluated (but perhaps that code is not optimized, or you didn't get to try every parameterization

[1] Anywhere from 10-15 minutes is OK.

you'd like, or you didn't get to run it on the truly desired input, or the evaluation is not quite thorough enough to declare a huge victory). If you pick a survey project, this might mean that you thoroughly detail progress, providing clear and engaging context, on a specific problem through a specific approach used in a small set of papers (but perhaps don't consider every paper that has addressed this problem).

For all projects, we can spend a brief meeting helping you get started, but the goal is for you to make your own concrete plan, and to complete the project independently.

How to approach the writeup. One ballpark to have in mind for a final writeup is ≈ 8 pages, but this is neither a ceiling nor a floor. Depending on your project, you may wind up using more pages for figures or precise definitions, or you may wind up using fewer pages. Please include exactly the content that you think is interesting (please do not cut something interesting out to keep it short, and please do not add filler material to make it longer — just try to make the writeup as engaging as possible). **We would like to post all final reports on the course webpage**, so you should write with high enough quality that you are comfortable having your name attached to the PDF online. We would like to do this (a) to give examples of projects for future iterations and (b) because some of the projects will be useful to others. If necessary, we'll make exceptions (e.g. because your project may turn into a paper and you're not ready to post partial progress online yet).

Please try to make the *substance* of the writeup high-quality. For example, you should have precise definitions, intuition to guide any proofs, and organized figures. Any context/background you provide should be clear and detailed. This will likely take you a few rounds of edits to get right. At the same time, please *do not spend effort* making the writeup "look polished." It's OK if you decide to do all equations in display math instead of align, if some margins run over, and if you don't "sell" the work in your introduction (you're not submitting this to a conference). But please still make the substance/content of the writeup technically engaging and easy to follow.

How to approach the presentation. Similarly, please make the substance/content of the presentation high-quality, but please do not spend effort polishing the talk. For example, it is completely fine if you write a_i (or $a.i$, etc.) on a slide instead of a_i. It is completely fine if your slides are text-heavy, as long as they get across the content clearly. But please put some thought into choosing exactly what content to present, how to present it clearly, and how to do so in your allotted time.