CS166 Handout 09
Spring 2018 May 1, 2018

The CS166 Final Project

The final project for CS166 will serve as a capstone for the course. In it, you'll do some independent reading and research into a topic of your choice, will contribute something new to the field, and will present your findings. In past iterations of CS166, students had a great time with these projects and we were truly impressed by what people came up with. We're excited to see what you as a group develop over the rest of the quarter!

Project Proposal Due Thursday, May 10th at 2:30PM.

Final Project Requirements

The final project will work as follows. In a team of two or three, you will choose a data structure, an algorithm pertaining to a specific data structure, or a theoretical result about data structures that will serve as the focus of your project. You'll then track down a research paper on your topic and become an expert on your particular topic by reading that paper and finding other sources (textbooks on the subject, lecture slides, follow-up papers, etc.). You’ll then put together a writeup of that data structure and present it to the course staff.

Having read up on the topic, you'll then do something “interesting” with it. You're welcome to do whatever you think is appropriate here. Here's a sampling of what students have done in the past:

- Implement the data structure, compare its performance against benchmarks or against theoretically-predicted behavior, and analyze the results.
- Research theoretical bounds on the performance of the data structure in the worst-case and formally prove those bounds.
- Apply the data structure in a novel context or to solve an existing problem in a new way.
- Write a simple, unified framework for analyzing the data structure and proving correctness.
- Modify the data structure using ideas from other data structures and analyze the practical runtime impact.
- Design a visualization of the data structure that makes it significantly easier to understand.
- Explore the progression of related algorithms and data structures and explain how the data structure fits into a broader context.
- Compare the data structure against several other related data structures to identify its strengths and weaknesses.
- Repeat a runtime analysis done in an (older) paper on modern hardware to see how well the results hold up.

We expect that the project will be two-thirds focused on understanding and presenting the original paper and one-third on the “interesting” component.

Deliverables

Your final project will have two deliverables.

First, we'd like you to prepare a writeup on the data structure. In your writeup, you should describe the data structure, explain how it works and any key results, and flesh out any relevant proofs. You have some latitude in how you produce your writeup. You could write it up as a standard technical report, aimed primarily at theory-minded folks. Or you could put together a long-form article (say, a detailed blog post) describing the data structure to practicing programmers with a solid math background. Either way, the goal here is to convey the data structure clearly and in a way that helps other people see what makes it tick and why it's so exciting. After all, if you've found a data structure that you think deserves broader adoption, you'd want to make it easy for people to see what it is and how to use it!

Your writeup will be evaluated both on correctness, demonstrated mastery of the topic, and clarity of presentation. Produce something that, stylistically, you'd be proud of – write in complete sentences, introduce section headings as appropriate, etc. Avoid speling or grammatically errors, as they will detract from the quality of your writeup. There is no minimum or maximum length requirement – write as much as is necessary to cover all the major points in a way that's clearly accessible.

You're welcome to share drafts of your writeup with other CS166 students to get feedback, and in fact you're encouraged to do so. Just make a note of it somewhere in your writeup.

The writeup is due 24 hours before your group gives its presentation. This will give us time to give a quick read of your writeup before you give your presentation. For example, if your group is presenting on Monday, June 4 at 3:30PM, your writeup is due on Sunday, June 3 at 3:30PM.

Please be sure to cite your sources. You are required to cite your sources in your writeups. If you copy or paraphrase any text, proofs, or figures from other sources, you need to cite that source so that we know where you got it from. It should be easy for us to differentiate between your own words and ideas and those from the sources you've read.

Your second deliverable is a 15- to 20-minute presentation that you'll give during the last week of class.

As with the writeup, you should use this time to introduce your data structure, describe it, and present the "interesting" component of your project. Unlike the writeup, during your presentation you do not need to give a complete description of your data structure. Instead, present the basics – what does it do? how does it work? what are the time bounds? – in a way that makes it accessible and understandable. Your job should be to communicate the data structure and present your "interesting" component in a way that makes the structure seem as interesting and exciting as possible.

At the end of your presentation, we'll open for questions from the course staff and anyone else present. The course staff might ask some brief questions on your presentation or data structure at this time, so you may want to quickly review your writeup and your data structure beforehand.

These presentations are open to the public. You can probably expect that two or three members of the course staff will be there, along with other CS166 students and generally interested members of the CS community. Accordingly, you can assume that the audience has a technical background at the level of CS166, but you should not assume that they already know what your data structure is or how it works. In particular, even though the course staff will have likely already taken a quick look at your writeup, we won't necessarily have a full command of the data structure at the time you present.

Because presentations are spaced apart at roughly 30-minute intervals and we want to have five minutes for questions at the end, we may have to cut off early any presentations that run over the time limit. Additionally, please arrive at least five minutes early so that you have time to get set up.

Overall, your grade on the final project will be based on three factors:

- ***Your demonstrated understanding of the topic at hand***. Your job in the final project is to become an expert on the topic that you've picked. Aim to *explain* rather than *repeat* information. Is there an unusual constant chosen somewhere in the data structure? Perturb it, see what happens, and come back with an explanation as to where that constant comes from. Is there a tricky algorithm that isn't immediately intuitive? See if you can come up with a good visualization for how it works, or come up with a novel explanation.
- ***The quality of your presentation, both written and oral***. Although data structures are highly technical, you should be able to communicate your data structure effectively. You can assume that your audience understands the material from CS166. This applies both to your paper and to your presentation. The best written reports are ones that make the data structure seem natural, intuitive, and interesting, and the best presentations are the ones that make us seriously considering changing the syllabus to include the topic in the next iteration of CS166.
- ***How "interesting" your addition is***. You don't need to go above and beyond with this step to earn a high grade on the final project, but what you choose should be somewhat interesting. Ideally, your "interesting" step should provide a richer understanding of the data structure or the context in which it resides. The best "interesting" components we've seen were so good that they were probably publishable.

The Final Project Proposal

To ensure that we don't end up with too many people covering the same topics in their final projects, your first step in working on the final project is to submit a list of project proposals.

First, determine who you'll be working with. You are required to work in a group of two or three unless you receive prior approval from the course staff (that is, we specifically authorize you to work individually *before* you submit the project proposal). This is partially because we expect the final project to be a lot of work and partially because we simply don't have enough time during our presentations schedule for everyone to present individually.

Next, we'd like you to list, in ranked order, ***five*** choices of topics for your final project. We'll give you a list of suggestions in another handout, but you're welcome to choose any data-structure-related topic you'd like as long as we haven't covered it this quarter and won't cover it later on. For each topic, we'd like you to track down at least one paper on the subject and at least one other source on it – it could be lecture notes from other courses, a book chapter, a blog, a follow-up paper, etc. – to ensure that you have a sense of where to look for more information. Additionally, for each topic, we'd like you to list one or two possible ideas for your "interesting" component. These ideas aren't binding, but rather are there to ensure that you've actually thought things through.

Once we've received everyone's submissions, we'll run a matchmaking algorithm to assign topics to teams. Once your topic is assigned, you will need explicit permission from the course staff to change it. This is mostly to ensure that we don't end up with duplicate topics.