

Final Project

Description

Final course projects count as 20% of your final grade, with an optional additional 10% extra credit according to the novelty, implementation, difficulty, and report writing. You can work on your final course project individually or in groups of no more than three students. More group members need to demonstrate more work in the final project.

Please use the link posted to Piazza to accept the final project and push all the materials to the created git repo. The deadline for the final project will be at 11:59pm EST, April 27th 2018. No submissions will be accepted after the given deadline. All the work needs to be submitted to Github. Submissions by email will not be accepted.

Proposal

Please submit a proposal for what you want to solve in your final project on Piazza. You can either list your group member. Or you can use this chance to find someone to work with you.

The proposal should be no more than 2 short paragraphs long. Your TA and your instructor will evaluate your proposal to check for the difficulty. Proposal due on Feb 23th, 2018. Everyone should have a team by then.

Submission

Your project submission consists of three parts:

1. A project report `report.md`, written in markdown format and should be directly viewable on Github. The whole report should be no more than 2000 words.
2. Push all your files to the Github repo. Any C++ code should be accompanied by a Makefile, and should compile with no errors on the c9.io workspace. If you are familiar with CMake, you can also organize your project with CMake. Other compilation methods will not be accepted. Also, make sure to give an instruction in a `readme.md` file about how to compile and run your code.
3. If you think a video helps your demonstration, you can record video no more than 5 minutes long and upload it to Youtube. You should paste a link to the youtube page in your report.

Report

Your project report should have the following sections:

1. Project title
2. Authors
3. Introduction

What problem you solve in your project. Why it is important. Motivate readers in this section without too many technical details.

4. Background (Optional)

If your project is based on others' work, briefly introduce that work. Introduce the theory and terminologies you are going to use in the next a few sections.

5. Design

What is your approach. Briefly go over the technical details and challenges of your implementation. What you have done to improve performance.

6. Performance

What is the overall performance of your design. How you interprets that. Do your performance optimization approaches work?

7. Relevance

How your project links to the topics we covered in the class.

8. YouTube link (optional)

Add here a clickable link to your YouTube video.

Project topics

Just in order to give you an idea of the complexity expected from your final project, here are some example projects:

- Efficient matrix multiplication algorithm implementation
- Solve a rubics cube with algorithm
- Solve a 15-puzzle problem with algorithm
- Use KMeans algorithm to cluster data
- Use PageRank to evaluate the importance of a node in a map
- Multi-threaded Monte-Carlo method for PI calculation
- Multi-threaded Merge Sort algorithm
- Implemtation of a simple neural network for classification

These are just some examples, and you are encouraged to come up with your own project ideas.