Algorithms for big data
Project
2021

Post questions to the project channel of teams

Project guidelines

Your grade is 50% final written exam and 50% project.

The goal of the project is for you to learn an algorithm not covered in the class, implement it and understand its analysis.

Projects may be completed alone or in a team of at most two people. Both members of the team will get an identical score.

The main deliverables are the code and a 20-minute-long video.

The project is due on May 5 at 11:59PM. You will lose 1% per hour of lateness.

The algorithm

You should choose an algorithm that is

- Relevant to the field of big data. You can double-check with with about whether anything is on-topic but asking in the project teams channel.
- Published in a reputable conference or journal in the past 10 years. Examples of reputable conferences where algorithms results appear are can be found here. Note that in theoretical computer science, conference publication is typically the main way one gains recognition for a result. Recent papers are also posted on arXiv for convenience, but arXiv has almost no editorial filters. After appearing at a conference a paper may also appear in a journal, but the absence of this step says nothing about the paper's quality.
- The algorithm must have a rigorous analysis.
- You have freedom of choice, you can pick something more general or some algorithm developed for a particular domain where big data is prevalent, e.g. bioinformatics.
- I should not be an author of the paper you have chosen.
- No duplicates are allowed. Each group should broadcast their choice on the Project teams thread. Whoever announces first has the right to submit the project on that paper. Duplicate projects who did not announce on teams will get a zero score.
- Don't choose something that has publicly available code.

The implementation

You should understand and implement the algorithm. You should test it to see that it works and that the theoretical claims about the algorithm (runtime, space) are true. You should try to compare it against other solutions, either publicly available ones, or "trivial" ones as appropriate.

You code should be in any widely used language.

You should convince me that the code is correct. This could be done graphically.

The video

You should create a 20 minute long video. In this video, you should explain the problem, indicate the model (e.g. streaming, cache-oblivious, etc), the algorithm, what is the theoretical performance of the algorithm, what was known before this algorithm. You should usually show how the algorithm works on an example. The goal is that after I watch the video, I should understand the algorithm and its analysis.

You should then present the results of your implementation (5 mins max).

To hand in

- Email to bigdata21@johniacono.com
- Subject: "Project submission of XXX (and XXX)"
- CC your partner, if a pair
- Attach a pdf of the paper
- Submit all files needed to run you code, including any data sets you may be using. If not using a notebook like Jupyter, include a README and a Makefile, where the README clearly says what I need to do to run your code as well as any other needed explanations and screenshots of your code running. If files are to large to attach, send a link.
- The video or a link to the video. Do not change the video after the deadline.
- I will confirm submission via email if you don't get a confirmation in a few hours during the working day, contact me via teams.

Questions

I may ask you questions via email after looking at your project to clarify. You will have two working days to respond to any email and these emails will be considered to be part of the project submission.

Grading

Grading is 50% implementation (including the part of the video where you discuss your implementation) and 50% video (excluding the implementation discussion).

The implementation score will be based on - The difficulty of the implementation. Note that the difficulty will be adjusted based on whether the project is done by one or two people. - The quality of the code, the documentation - How well the code implements the algorithms in terms of correctness and metrics such as speed and time

The video score will be based on how well you understand the algorithm, its analysis and its historical context, and how well you can communicate these things. Difficulty will also play a role in the scoring.

Academic honesty

All work should be your own or referenced. Reference anything that is not yours. If you copy more than a single line of code from somewhere else include a comment with a URL. If you use an figure in your presentation that you did not make, it must be referenced. This does not mean that copying code is forbidden for some subroutine incidental to your project, it is not! Just give recognition where it is due. Any breach of academic honestly will result in a zero grade for the project.