Final Exam: 2 Short Essays on Algorithms

Friday, 16 December 2022, 3:15–5:15 pm, Chapman 206

The in-class Final Exam will be a bit different, but relatively short. I would like you to write two essays, each with length in the range from 200 to 300 words, on certain optimization algorithms we have covered, or on important algorithm components. These are topics we *have* covered in lecture and homework, and this is an exam not a project.

In particular, please

choose 2 topics from the 7 topics listed at the bottom,

and write a 200–300 word essay on a separate sheet for each of your two topics. Make sure to read the indicated sections of the textbook for your topics.

You may not bring notes to the final, but you are **strongly encouraged to practice** your planned essays. Feel free to get feedback on your practice essays from other students or faculty/friends/family/pets. Then think through, in preparation, how you will remember enough detail so as to recreate the essay during the Final Exam itself.

Note that 200 words equals 20 lines of text with 10 words on each line, or 30 lines with 6.7 words each. Thus each essay should be less than a page, though perhaps close to a page if you write large. Use equations as appropriate to communicate meaning, but not more than a few, and likewise one or two figures as needed. Most of your essays should be words.

Your goal is to write something like a short wikipedia entry on the topic. For example, if someone searches `line search` in google, what they should get on the first two screens of the Wikipedia entry might be something like your essay for topic (4) below. Said a different way, if a friend asks you to explain what a "line search" is, and why you want one, what would be your 5 minute answer? A good 5 minute oral answer, perhaps augmented with scribbling on a whiteboard, should be converted here into a clean, clear, short essay.

For reference, the three paragraphs just above this sentence total 218 words.

The 7 Topics.

(1) How to move around a feasible set defined by linear equality and inequality constraints, including the ratio test. *(section 3.1, especially pages 80, 81)*

(2) How to put linear programming (LP) problems in standard form. *(section 4.2)*

(3) The simplex method for LP problems in standard form. *(sections 5.2–5.4)*

(4) Back-tracking line search to generate sufficient decrease. *(section 11.5)*

(5) Steepest descent using exact line search on quadratic functions. *(section 12.2)*

(6) Quasi-Newton methods, with symmetric rank-one as the example. *(section 12.3)*

(7) Optimality conditions for minimization subject to *linear* equality and inequality constraints. *(sections 14.3-14.4)*