CMPSC 250 – Analysis of Algorithms (Spring 2015)
Prof. John Wenskovitch
http://csALLEGHENY.EDU/~jwenskovitch/teaching/CMPSC250

Course Project
Due (via Bitbucket and hard copy, plus presentation) Tuesday, 28 April 2015
100 points

Lab Goals

• Create and evaluate your own algorithm(s)

• Solve a real-life problem or develop a real-life application

Project Details

A significant portion of your final course grade is the project component. For your project, you will work in groups of three or four (preferably), but each member of the group will be evaluated separately based on his or her contributions to the project. This evaluation will be determined largely from the feedback of the other group members. You may also choose to work on your own if your schedule will not allow you to meet with other group members regularly. However, I would discourage this option.

In your project, you are to design a real-life application system that uses a new algorithm or a series of new algorithms. You have a significant amount of flexibility with what you can do with your project, so use it wisely! Is there something you wanted to create but never had time? Is there a project from other courses that you always wanted to extend? Juniors: is there an idea you have for your thesis that you want to try out first before committing to it? Seniors: did you want to explore a different aspect of your senior thesis? If the answer is “yes” to any of these, and if you can connect your idea to algorithms, then you can do it as your project now! Of course, you cannot claim anything you have previously done as a contribution for this project, but you can certainly use your previous work, knowledge, and experience as a backbone for this project.

Hard Requirements

1. Your project must be approved by me before you start working on it.

2. Your project must solve a real-life problem or develop a real-life application. You need to research the problem you select to get an idea of what has already been done. You must include references to existing work in your final report.
3. Your project must involve designing new algorithm(s). You can approach this requirement by either:

- Creating a brand new algorithm and applying it to some problem. You must do some research to make sure your algorithm has not been proposed before.
- Designing a significant extension to an algorithm and applying it to some problem.
- Most likely, you will need to develop or extend a series of several algorithms.

In any of these cases, you must provide a discussion (or proof) on the correctness of your algorithm. This discussion can include small examples and references to existing work (especially if you are extending existing algorithms).

4. You must provide a step-by-step mathematical analysis of the running time of your algorithm(s) and its components. Mathematical analysis should include a complete mathematical derivation for the running time of the algorithm, starting with each component of the algorithm and ending with a closed form mathematical expression. The running time (∼ notation, average case) should then be determined from this mathematical expression.

5. Your project must have a significant implementation part where you will develop program(s) for your algorithm(s) for your chosen problem. You may write your code from scratch, or reuse and extend some existing code. Obviously, anything you use that is not yours must be documented. You may program in any programming language that you like.

6. Your project must be extensive enough to qualify as a project (think of work for at least 5 one-week lab assignments), but not too extensive so that you cannot finish it in the reminder of the semester (two months).

**Project Examples**

Here are some examples of possible projects:

- Class Scheduling – Help your fellow Allegheny students schedule their classes.
- Housing Assignment – Develop a system that will “fairly” assign housing to students, taking into account the rules used by the college now.
- Art – Develop automated music or art creating algorithms.
- Security – There are a lot of algorithms you can develop to address cybersecurity issues. For example, you can develop algorithms to forecast cyber attacks or software vulnerability.

There are many references to all of these problems, and I’m sure as you think of your own project, you will find resources for them as well. You may use any of my example problems if you can’t think of anything in particular yourself. Be creative and choose something that is interesting to you!
Timeline and Deliverables

1. Proposal – 1-2 pages, Deadline March 9

   Develop an idea for your project. Write a 1-2 page technical description of what you propose to do for your project and submit a paper copy to me sometime on Monday. (Storing a copy in a BitBucket repository dedicated to the project would also be a good idea.) Your proposal does not need to be very detailed at this point. It should describe what you want to do (the real problem you will tackle and the type of algorithm(s) you will develop), and at least a couple of references to show me that you have done some research about the problem. But you do not need to include any specifications on how you will solve your proposed problem. That is, you do not need to include any details about the algorithm(s) you will develop.

   Meet with me during the lab period on March 11, or schedule an appointment during my office hours during the week of March 10-13 (I want to save March 9 to review all proposals before these meetings) to discuss your proposal and to receive an approval or a modification to your project proposal.

2. Progress Report 1 – 2-3 pages, Deadline March 30

   By this point, you should have designed a model for your system, or a framework for solving the proposed problem; developed your algorithm(s), written them in pseudocode; and started analyzing the complexity. Also, you should have started programming.

   Include everything that you have done so far in your progress report, even if it is incomplete.

3. Progress Report 2 – 3-4 pages, Deadline April 13

   By this point, you should have made a tremendous amount of progress towards implementing the algorithm(s). Were there any unexpected challenges? Did you have to change your initial model/framework or the pseudocode? You should have also finished your mathematical analysis of the complexity by now.

   Include everything you have done so far in your progress report, even if it is incomplete. No need to include the actual code (unless you want my help with it), just describe what progress you have made with it.

4. Presentation – Deadline April 28, possibly also April 23 depending on the number of groups

   By the presentation session, you should have finished implementation, run some preliminary experiments, and done some basic analysis. In the presentation, you should describe the motivation, problem definition, challenges, approaches, and results and analysis. Use diagrams and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Prepare for a ~10 minute presentation. Design at least 5 slides, including a slide with the title of your project and group members.

   Every member of the group needs to contribute to the presentation talk. Also, you need to show a demo at the end of the presentation of your system.
5. Final Report – 7-10 pages, Deadline April 28

Incorporate any feedback from the progress reports and the presentation session. Your final report should be clear and well written, which includes no typos or grammatical errors. The writing will be graded more harshly here. Your report should be written in a professional and technical manner. Your report should include the following:

- The motivation for your project. Why is the problem you decided to solve important or useful?
- Background for the proposed problem. What have others done for it already? Include references.
- Detailed algorithm description and the complexity analysis. Include pseudocode, diagrams, and examples if appropriate. If you are extending existing algorithm(s), briefly describe previous work and include references to it.
- Description of your results. Make graphs, tables, and anything else that can help me understand your results.
- Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges and the biggest rewards.

Submission Details

For each deliverable, you need to submit a PDF with your report (or presentation slides). For your final report, you need to submit any supplementary material (code, data, a README file documenting what everything is, and how to run your program) to BitBucket.

- Proposal – 10 points
- First progress report – 15 points
- Second progress report – 15 points
- Presentation – 25 points
- Final report and implementation – 35 points

Please remember that all files that you submit should be your own work, though you are welcome to discuss high-level topics and algorithms with classmates.