Project: Applying Discrete Mathematics to Computer Science

Overview
As the capstone project of this course, you will apply discrete mathematics to solve a practical computer science problem of your choice and to write up your work in a brief technical report. In particular, you must state a focused practical problem, model the problem mathematically, solve the problem using the techniques of discrete mathematics, and interpret your solution in the original practical domain. Your project must have some clearly identified original component—literature surveys and paper reviews are not acceptable.

As an alternative project, you may prepare a learning package on any topic in discrete mathematics.

Due dates
Proposal: Monday March 7.
Progress Report: Monday April 11.
Abstract Page: Monday May 2.
Complete Final Report: Monday May 9.

All items are due at the beginning of class (*i.e.* at 4:00pm in SS 206) on the day they are due. Late reports will be penalized by one letter grade for each 24 hours of lateness and will not be accepted after 5:15pm Friday May 13. The rest of this document describes in detail what is required for each of these four items.

Purpose
The main purpose of the project is to give you an opportunity to apply discrete mathematics to solve a practical computer science problem that fascinates you. Whereas most homework exercises are limited in scope and can be solved within a few hours, the project gives you an interesting open-ended problem to think about over the entire semester. In addition, doing the project will help you learn how to carry out research and how to communicate your findings in writing.

How to pick a problem
To pick a problem it is helpful to consider the following process. First, select a topic that fascinates you (*e.g.* cryptology, algorithms, artificial intelligence, databases, graphics, computer music). Second, decide on what type of skill you wish to focus (*e.g.* counting, solving recurrences, proving theorems, estimating values). Third, find a practical problem in your topic that can be solved by your skill (*e.g.* count the number of possible keys in the Geheimschreiber cipher machine). Be sure to keep your problem highly focused and within your grasp.

Selecting a problem to solve is one of the most difficult steps in carrying out this project. Many students find it helpful to discuss their problem ideas with the instructor early in the semester. It is highly desirable to draw upon your life experiences in selecting your topic.

Although the instructor prefers each student to select her own problem, the following list of sample projects may help you get started selecting your own problem.

1. Analyze the time and space complexity of an algorithm by setting up and solving a summation or recurrence. A good source of new algorithms to analyze is any non-mathematically oriented paper that vaguely describes some applied algorithm (often with such papers, part of the problem is to determine what is the algorithm).

2. Count the number of objects of some interesting type. For example, count the number of keys in various cryptosystems, or count the number of data structures of a certain type.

3. Estimate an important value. For example, estimate the key lengths required of various cryptosystems to achieve various levels of security for various numbers of years into the future. Or estimate the probabilities of various events.

4. State and prove a theorem. A good source of theorems to prove is any applied paper that does not rigorously justify its claims. Another strategy is to prove a special case of an unproven theorem.

5. Model the behavior of some process. For example, model the hardware failures in a supercomputer under various situations.

In spring 1992, student projects included the following: modeling the spread of computer viruses with difference equations, using fractals to generate Japanese flute melodies, solving hand ciphers with Boolean equations, estimating the time required to defeat the new Digital Signature Algorithm, and preparing a learning package on how to solve recurrences with generating functions.

What to hand in

You must hand in four items: a proposal, a progress report, an abstract page, and a final report.

- *Proposal.* Briefly state what you plan to do and how you plan to it. Your proposal must contain the following items:

1. Descriptive title.

2. Brief statement of practical problem to be solved.

3. Brief explanation of why it is important to solve this problem.

4. List of specific, well-defined questions to be answered. *These questions are the most important part of your proposal.*

5. Brief explanation of what methods you plan to use to answer the aforementioned questions (*e.g.* formal proof techniques, combinatorics, difference equations, estimation, mathematical modeling).

6. Realistic schedule for completing your work.

7. List of any difficulties you foresee.

Limit your proposal to at most five pages.

• *Progress Report.* Describe your progress to date. Your progress report must contain the following items:

1. Descriptive title.
2. Introduction. The introduction must identify what problem you are solving, why it is important to solve this problem, and what is new about your work. In addition, the introduction must identify specific, well-defined questions to be answered.
3. Background. Carefully review selected relevant background material necessary to understand your work.
4. Detailed outline of paper.
5. Current status. Explain what you have done to date and what remains to be done. Identify any potential difficulties you foresee.
6. Bibliography. List all sources you plan to use.

• *Abstract Page.* The abstract page is a one-page summary of your work (for an example, see the "CMSC-693 Guidelines"). Abstract pages are due one week before classes end so that other students can learn about your work. The abstract page must contain the following items:

1. Descriptive title.[1]
2. Author name and affiliation.
3. Date.
4. Brief informative abstract that summarizes the original scientific content of your work.[2]
5. List of keywords.

• *Final Report.* In the format of a brief technical report, communicate your work in writing. Explain what you did, why you did it, what you found, and the significance of your findings. Clearly state and answer the questions you addressed.

For advice on how to write a technical report, see the document "CMSC-693 Guidelines."[3] For additional advice on technical writing, see the handout "How to solve and write up homework problems."

[1] The title "CMSC-603 course project" is not descriptive.

[2] Discussing what you *addressed*—as opposed to what you *found*—is not informative. Specifically state your concrete results.

[3] Do not include a formal CMSC-693 style title page.

Be concise; do not pad with fluff; and concentrate on the central scientific issues raised by your questions. Emphasize what is new about your project. Present your findings and discuss their significance and limitations; do not simply state your findings. Make sure your report is clear, informative, well-organized, and easy to read. Take as your audience another student in this class.

To encourage you to be concise and to focus on important scientific issues, you should strive to keep the body of your report at most six pages long. The body of the report must not exceed ten pages.

The instructor recommends that you prepare your report using a mathematically-oriented document preparation system.[4] Doing so will enable you to edit your work easily and to produce high-quality typography.

Your report must contain the following items:

1. Abstract page.
2. Body of report. The body must include the following sections: introduction (description of practical problem and motivation for solving problem), mathematical model of problem, solution, and discussion.
3. Acknowledgments.
4. References.
5. Appendices, if any. (For example, if you do any computer work, you must include a documented listing of all source code.)

Grading

Your project grade will be based primarily on your final report and will count for 20% of your semester grade. To pass the course, it is necessary to hand in a final report and to make a serious effort to complete the project.

This report will be evaluated according to the standard three criteria for computer science reports: *appropriateness for assignment*, *scientific merit*, and *effective presentation*. A report is appropriate if it meets the letter and spirit of the project assignment. A report has high scientific merit if it is correct, significant, novel, based on sound methods, nontrivial, and scientifically complete. A report is presented effectively if it is clear, well-organized, easy to read, complete in document, written in standard American technical English, and free of errors in grammar, spelling, punctuation, and typography.

The grade will be based on quality—not quantity. It is much better to solve a manageable focused problem than to attempt without success to solve an unwieldy broad problem.

How to earn a low grade

To earn a low grade, repeat each of the following common mistakes:

[4] The instructor recommends that you use the Gnu Emacs text editor and the LaTeX document preparation system.

1. Ignore all project instructions.
2. Wait until the last possible moment—preferably the evening before the report is due—to begin work.
3. Do not explain what problem you are solving, why you are solving it, or what is your solution. Most importantly, do not say what is novel about your work.
4. Omit all discussion of the significance of your findings.
5. Leave out as many required sections of the report as possible, especially the abstract, list of keywords, acknowledgments, and references.
6. Begin your abstract with the hackneyed phrase "This paper ...", and make certain that the abstract is as uninformative as possible. An especially effective way to write uninformative abstracts is to describe vaguely what problem you addressed rather than to report specifically what concrete results you discovered.
7. Neither define, typeify, nor quantity any free variable.
8. Be sure that no graph axis or table heading is labeled.
9. Do not bother to edit or proof read your report.

Alternative projects

Although the instructor would prefer you to carry out this project as described, if this project does not excite you, *with prior special permission of the instructor*, you may carry out a different style of project. It is important that you are excited about your project.

For example, you may do one of the following alternate projects.

1. Prepare a "learning package" on a focused topic of discrete math of your choice to be used by future CMSC-603 students. For example, you could prepare a learning package on how to use generating functions to solve recurrences. The package must include the following items: written explanation, suggested readings, novel exercises and model solutions, a video-taped presentation, and a brief description of the goals and methods of your package. I highly recommend that you present your package using the available hypermedia resources at UMBC.
2. Design and implement an expert system for solving a focused type of problem of your choice. For example, you could design and implement a system to solve summations, recurrences, or Diophantine equations symbolically. Your report must include a detailed user's manual, a demonstration of the effectiveness of your program, and a discussion of the engineering issues in your design.
3. Write a research paper on the history of the solution of an important problem in discrete mathematics of your choice. For example, you could research the exciting history of Hieronymo Cardano's (1501–1576) solution of the cubic equation $x^3 + ax^2 + bx + c = 0$. Your report must focus on technical details and be based in part on primary sources.