Final Project Instructions
MATH 3012 QH: Applied Combinatorics, Fall 2020

There are five parts to the project. Students must work in a team of two or three people. The final page of this document provides a list of potential projects.

1 Due Dates

- Part 1: 10/15, 11:00 pm, Thursday
- Part 2: 11/05, 11:00 pm, Thursday
- Part 3: 11/16, 11:00 pm, Monday
- Part 4A: 11/19, 11:00 pm, Thursday
- Part 4B: 11/20, 11:00 pm, Friday
- Part 5: 11/20, 11:00 pm, Friday

2 Instructions for each Part

2.1 Part 1: Team Up and Select a Topic (5 points)

Students must indicate which topic they would like to pursue and who they are working with.

- To select a topic, your group must write a note on our class Piazza forum that states who is in your group and what topic you have chosen. Write a few sentences about what your topic is about.
- Tag your post with the ”project” folder in Piazza.
- Your post should be set as a note (that doesn't require a response), not as a question.
- One person in each team submits this information.
- All projects are approved automatically, unless the instructor indicates otherwise.
- Students who do not have anyone to work with are encouraged to write a post on Piazza to say they are looking for someone to work with.

2.2 Part 2: Report Draft (10 points)

The purpose of the Report draft is to give an update on your work, encourage students to not leave work to the last minute, and give the instructor an opportunity to give feedback on student work. This part of the project is only graded for completion. Your instructor may offer feedback or suggestions at this stage.

- Each group should submit one Report as a PDF document.
- The Report text should have a title, and be organized into sections with appropriate titles. Your should have the following components.

  - A title of the project at the top of the document.
  - Names of the students in your group (also at the top of the document).
  - A brief **abstract** that summarizes the problem you are working on, your approach, and what you found. Roughly 100 to 150 words. Located after student names and before introduction.
  - An **introduction** that introduces the topic or problem, states why it is of potential importance.
  - A **conclusions** section that summarizes what you learned.
  - Other sections that develop the main ideas of your work.
  - A references section with at least three references (you can cite the textbook as one source).

- This part is only graded for completion.
- Please submit your draft through Gradescope.
- The draft should be roughly 1000 to 1500 words, not including code (if any) and references.
- Diagrams, tables, and figures (if any) should be described in the text. Do not assume the reader will know what to infer from these elements.
- All variables in the equations that you present need to be defined.
- The report must be typed. Spacing can be single or double, the font can be 11 pt or 12 pt.
- If you wrote a computer program or code, please submit their source code. The code can be added as an appendix.

2.3 Part 3: Report Draft for Mini-Conference (10 points)

Submit a draft of your report on Piazza. Please include all of the components listed in Part 2. Students may be asked questions from the instructor and/or other students. Students must share their Report on Piazza as a note and tag it with the project folder.

2.3.1 Style

Students do not need to use a particular style for formatting their references, citations, headers, and so on. It would go beyond the scope of this course to ask that students learn and use a particular style. But students are welcome to use any style that they prefer, as long as they use a consistent format. Information in the citations must be correct. If anyone would like to use a particular style, they may want to use one of the following style guides when formatting their work.

- The American Mathematical Society Style Guide:
  www.ams.org/publications/authors/AMS-StyleGuide-online.pdf
- The American Psychological Association Style
  www.apastyle.org

2.4 Part 4: Virtual Mini-Conference (10 points)

This part of the project is only graded for completion, not for accuracy. Please read other Reports and comment on them using the guidelines below. Please also answer questions that students pose.

- **(6 points) Part A**
  You should ask at least one question for two other groups. In other words, pick two different Reports, and ask at least one question for each Report. Each person in your group has to do this, so this part is graded individually. Of course, don't ask your own group a question! Please also, when asking your question:
  - write something you liked about the draft you read, or something that you learned from reading it
  - state whether you found the work to be understandable - and if it wasn't, ask for clarity on an area that you found confusing
- **(4 points) Part B**
  Each person in your group receives full points if all the questions that your group was asked about your work were answered, or at least four questions were answered.

2.5 Part 5: Submit Final Poster (25 points)

You will submit your final draft of your Report through Gradescope after the conference. Students are encouraged to incorporate feedback they receive during the conference before submitting their final Report. Please include all of the components listed in Parts 2, and 3, in addition to the following.

1. Background: What problem are you solving? What references did you use to learn about this? Please cite references.
2. Solution: Your work should propose at least one mathematical model, algorithm, or approach to the problem you are exploring. Explain how you selected it, and what assumptions your group made (if any), and why you made them. What limitations might your approach have?
3. Concepts: describe concepts related to the course and made explicit reference to them. Please cite our textbook. If you are working with different mathematical models, compare them.
4. Accuracy: free of mathematical errors
5. Formatting: free of grammar and spelling errors, and typed.
6. Novelty: explored areas that were not covered in the course. For example, you may want to use techniques that are in our textbook but were not covered in our course. If you collected data or simulated data, how did you gather/calculate the data?
7. References: include a references section with at least three references that are not websites, one of which can be the course textbook.

If you wrote a computer program or code, please include your code as a separate file containing the source code.

3 Potential Project Topics

What follows is a list of possible topics that are short descriptions of past student projects.

1. A computer program that can automatically draw out a trail for someone to take for a walk, based off of data from open street map, and ideas from this class. For example, to avoid going on the same path twice.

2. In Lecture 10, Trotter went over the history regarding Berge's perfect graph conjecture. He specifically mentioned that an important result towards the conjecture was found in 1972, and then 4 mathematicians (including one from GA Tech) found the proof in 2006. Explore this proof.

3. A coding project that involves the solving of scheduling problems using graph implementations.

4. In Robotics, the first 30 seconds of competition are dedicated to autonomous driving - aka driverless control. Robots attempt to gain as many points possible by scoring game pieces, moving spaces, etc. One of the major uses of graph theory is in resource allocation. Give analysis of different possibilities for different robots by using weighted graphs.

5. Explore three proofs on how there exist triangle-free graphs with any arbitrarily large chromatic number shown in lecture 9, i.e. the pigeonhole principle proof, the Mycielski construction, and the shift graph proof. Further explain the proofs and create code to go along with the concepts for calculating how fast the graph grows in each construction.

6. Study the intersections of Game Theory and Political Science, specifically on using mathematics to chart opinions of the median voter and using that to mathematically predict an elected politician's stance on an issue.

7. Explore electoral gerrymandering and graph theory. Investigate ways to draw fair districts based on measures of compactness.

8. Investigate dangers of unsalted password storage and how major past breaches have happened as a result of storing passwords in plaintext or with a simple hash function. Explore how unsalted passwords can be cracked by creating a list of potential passwords ranging from easy (password123) to difficult (a random string of characters), attempting to crack their representative hashes in a variety of algorithms, analyzing their various completion times on multiple machines.

9. Investigate how one might characterize optimal paths for spreading information using the least possible number of re-posts on Instagram given a set of account data.

10. Study the relationship between combinatorics and game theory by examining the strategies involved in the board game called Go.

11. Techniques of combinatorial molecular biology with regard to their application in protein interactions. Specifically, sequencing and permutations of sequences associated with protein structures.

12. Adversarial networks, their connection to combinatorics, and their various applications such as the multi-armed bandit problem, parameter optimization, and feature selection. Utilize graph theory topics such as greedy algorithms and shortest path length.

13. Compare and contrast different path finding algorithms, including but not limited to Dijkstra's algorithm, Breadth-first search, Depth-first search and Uniform cost search. Discuss characteristics of the algorithms such as completeness (ability to find a solution if it exists), and optimality of the algorithm. We also may code programs to explain these algorithms in Python or some other language.

14. High schools in GA can have thousands of students. Scheduling class times at such schools is a challenging task. With large student bodies, mistakes seem inevitable. Seniors are enrolled in freshman classes, and students are missing courses on their schedule. Over-enrollment is also a major issue. Write a computer program that can model a more streamlined scheduling system for large high schools. Apply what you learn about graph theory in MATH 3012 to your code. Consider using graph coloring, network flows, and bipartite graphs, among other techniques used in 3012. Gain not only a better understanding of how combinatorics applies to real life but also a deeper appreciation for the teachers who have to schedule classes.

15. Write Java or Python software that shows some of the applications of Hamiltonian cycles to a simplistic yet common application, such as bus routing. School systems are given a large set of neighborhoods that are all in some geographical relativity to each other, and their goal is to find a bus route that is able to get through all the bus stops without making unnecessary stops and turns (for example, a bus going back and forth between neighborhoods might not be the best idea, when it is possible to go through neighborhoods more efficiently, only one time). Your project would further study the applications of Hamiltonian paths, which are paths in which every node/vertex is visited only once. If this path is a closed loop, then it is called a Hamiltonian Cycle.

16. Explore current methods of pathfinding for Euler circuits, from the greedy algorithm taught in class to Google's breadth first search algorithm, to develop your own algorithm for finding Euler circuits.

17. Explore the widely used RSA algorithm which encodes and decodes messages sent over the internet (also known as public key asymmetric encryption). Investigate how the difficulty level of RSA scales as you begin to use more bits, and compare that to the rate of growth of the processing power of modern day computers. Program various algorithms which are designed to crack RSA and run them on computers of differing power to see how long it would take a normal computer to crack RSA. Explore the idea of RSA being cracked by a theoretical quantum-computer running Shor's algorithm, a recursive algorithm that is currently believed to be able to break RSA given it can be run on a computer that utilizes qubits (quantum bits) instead of normal bits.

18. Explore the general topic of Combinatorial Chemistry with its applications in chemical processes such as molecule synthesis and pharmaceutical drug creation. However, more specifically within the realm of combinatorics' intersections with chemistry, explore the topics of using graphs to model chemical molecules and processes, as well as breakthroughs in ways of diagramming molecules.

19. Use graph theory - in particular, directed and weighted graphs - to model regulatory relationships between human genes. Genes often have effects on the transcription and expression of other genes; for example, the GTF2A1 gene in eukaryotes produces a transcription factor that amplifies the expression of other genes, a relationship which can be modeled as a directed graph from a vertex representing GTF2A1 to the influenced genes. Specifically apply

this directed-graph theory of genomic expression specifically to the genes involved in human embryonic development.

20. Explore planar graphs, with the implementation of a program to check if a graph is planar and to generate planar graphs.

Students can choose something similar to one of the ideas above, but are encouraged to propose their own ideas. However, all topics must be approved by the instructor and be related to the content of Math 3012.