Project
Due: Tuesday, April 11, 12:30 PM
Points Possible: 100

Overview.
This project is meant to give you a chance to more fully apply some of the concepts covered in the course and/or to allow you to explore some additional algorithms or data structures that are not covered in the course.

General Requirements

- This is not an individual project. You are required to work in groups of 2 or 3 people. If you need help finding people to work with, you can post notes on Piazza or ask for help facilitating.
- You have two main options for this project: **Option 1** is to do some research into something in the realm of data structures and algorithms that is not already covered in the course. I provide a list of suggestions below. **Option 2** is to design and implement a project that utilizes (in a useful way) several of the algorithms and/or data structures that we do cover in this class. Both options will require a programming portion and a presentation portion. This is not an in-class presentation and the mode of presentation can vary (see the details below).
- **Group Email:** You must email me (Lotz) with the names of your group members by Thursday, March 30 at 12:30 PM so that I can set up the submission groups on D2L before the deadline. This is worth 5 points of your project grade.

Specific Requirements for Option 1

- For this option, you will need to do research into a data structure and/or algorithm that is not covered in this class. **Please note also that if you have an honors contract for this course, you cannot choose the same thing you are doing for that.**
- For the presentation portion, you will present your research into the topic, explaining the data structure/algorithm in detail. You should include relevant discussions of implementations, space usage, and runtimes. Your presentation can be in the form of a *paper*, a *standalone slide presentation,* or a *short video.* You must include any references you used.
- For the programming portion, you must provide an implementation of the data structure/algorithm *from scratch*. All coding must be your own work. However, I will allow you to use any programming language you choose.
- Your programming portion must also include test or client code that illustrates clearly that your code works as expected.
- Your programming portion must include a README that explains what the code does and how to run it. Code should also be well-formatted and well-documented.
- Here are some suggestions for topics. **If you have a different topic in mind, you may submit it to me (Lotz) for approval, but it must be approved.**
    - Hybrid Sorting Methods–you should include at least two
    - Fibonacci Heaps

- Scheduling Problem
- Pattern Matching
- Regular Expressions & FSA
- Tries
- AVL Trees
- Huffman Encoding
- NP-Completeness–you should choose at least one NP-Complete problem
- Bellman-Ford Shortest Path

Specific Requirements for Option 2

- For this option, you will need to come up with a reasonably large programming project that uses at least three of the data structures and/or algorithms that we cover in class. Please note that these may not be used in a superficial way. You need to really think about their purpose and what they are for.
- For the presentation portion, you will present your project design, discussing the goal of the program, why you chose the data structures and algorithms, and what the space use and runtime is for the major pieces of code. You can present this in the form of a *paper*, a *standalone slide presentation,* or a *short video.*
- For the programming portion, you must provide a full implementation of the program. The three data structures/algorithms that you are using as part of the requirements *must be implemented by you from scratch*. However, you are allowed to use any programming language you choose.
- Your programming portion must also include test or client code that illustrates clearly that your code works as expected.
- Your programming portion must include a README that explains what the code does and how to run it. Code should also be well-formatted and well-documented.
- Feel free to run your ideas by me before starting on implementation.

Grading

Item	**Points**	**Some Criteria**
Group Email	5	emailed group member names on time
Topic Choice	15	complexity–the grade here can vary depending on whether you choose a very simple topic or something more complex
Presentation	25	● completeness ● clarity ● accuracy

		● details–i.e. doesn't have too many typos, etc.
Code Completeness	20	● README included ● test/client code included ● all required aspects included ● instructions for running and explanation of code included
Code Quality	35	● works as promised ● well-formatted ● well-documented

Note: I realize that not everyone likes working with other people. Due to the size of this class, though, we really cannot accept individual projects. Furthermore, this field is becoming increasingly team-oriented, so it's a good idea to learn how to work well with other people. This means that you are expected to pull your weight. If you have any issues with group members and would like guidance in working through it, feel free to reach out to me. If it turns out that a group member does not pull their weight during the project, I may assign different grades to different group members, but I will need to know the details of what is happening and what you have tried to do to mitigate the problem.

Submission.
These projects will be submitted on D2L. They may be submitted up to 3 days late, but there will be a 10-point deduction per day.