CSCI 230 Data Structures

Contents

1 Logistics **1**
- 1.1 Instructional Staff . . . 1
- 1.2 Key Dates . . . 2
- 1.3 Course Website . . . 2
- 1.4 Lecture . . . 2
- 1.5 Office Hours . . . 2

2 Course Description **2**
- 2.1 Prerequisites . . . 2
- 2.2 Workload . . . 2
- 2.3 Course Content . . . 2
- 2.4 Learning Objectives . . . 3
- 2.5 Course Text . . . 3

3 Course Structure and Grading **3**
- 3.1 Written Homework . . . 4
- 3.2 Programming Assignments . . . 5
- 3.3 Quizzes . . . 5
- 3.4 Engagement . . . 6
- 3.5 Cutoffs . . . 6
- 3.6 Regrade Requests . . . 7
- 3.7 Honor Code . . . 7

4 Course Policies **7**
- 4.1 Office Hours: Norms and Expectations . . . 7
- 4.2 Late Work . . . 8
- 4.3 Late Enrollments . . . 8
- 4.4 Attendance . . . 8
- 4.5 Modifications to the Syllabus . . . 8
- 4.6 Student Feedback . . . 9

5 Required Syllabus Statements **9**
- 5.1 Religious Holidays . . . 9
- 5.2 Students with Disabilities . . . 9
- 5.3 Inclement Weather, Pandemic or Substantial Interruption of Instruction . . . 9

6 Schedule (Tentative) **10**

1 Logistics

1.1 Instructional Staff

Instructor: Michael Levet (He/Him/His); lastnamefirstinitial (at) cofc (dot) edu.

1.2 Key Dates

Last Day to Drop Before Grade of 'W' Is Recorded: Monday, August 28.
Last Day to Drop Before with Grade of 'W': Friday, October 27.
Breaks: October 16-17 (Fall Break); November 22-26 (Thanksgiving Break).

1.3 Course Website

All announcements will be posted to the course website: `https://michaellevet.github.io/F23/CSCI230/index.html`. Students are responsible for checking the course website daily. Assignments and other course materials will be posted to OAKS.

1.4 Lecture

Lectures:

- TR: 5:00-6:15, Harbor Walk West 217.

1.5 Office Hours

There will be a mix of in-person (HWEA 312) and Zoom office hours. The Zoom link and days/times for office hours will be posted to my course homepage. Your success is my top priority- if any of these times don't work, please do not hesitate to email me to schedule an appointment! **If you have COVID or another contagious illness, please do not attend my office hours in-person. I will be happy to facilitate remote participation.**

2 Course Description

2.1 Prerequisites

The course prerequisites involve (i) two semesters of computer programming, with at least one of these emphasizing object-oriented programming in the Java language, as well as (ii) some flavor of Discrete Math. On the programming side, students entering this course should be able to independently design and implement algorithmic solutions in the Java language, debugging their code, and be comfortable consulting the Java documentation regarding built-in libraries. On the Discrete Math side, students should be familiar with first-order logic, evaluating summations (series), manipulating expressions involving exponentials and logarithms, and some proficiency in analyzing algorithms. While students will not be expected to formulate mathematical proofs, this course is mathematical in nature. Comfort with mathematical formalisms will be helpful.

Officially, the course prerequisites are as follows.

- CSCI 221 Computer Programming II (Grade of C- or Better)
- Math 207 Discrete Structures I (Grade of C- or Better)

If you have taken Math 295 Intro to Abstract Mathematics and wish to use that in place of Math 207, please talk to me.

2.2 Workload

CSCI 230 is a 3-credit course. Well-prepared students should expect to spend on average 9-12 hours/week outside of class. Students who have significant gaps in their backgrounds may find that they need to carve out additional time to review the prerequisite material.

2.3 Course Content

A *data structure* is a construct that is used to organize data, in such a way that is usable by other software components. The key operations to interact with a data structure are *inserting* new data, *searching* for existing data, *updating* existing data, and *removing* existing data. The way in which the data will be used often influences the design and choice of given data structure. In particular, certain data structures are often

useful for certain operations, or special cases theoref (e.g., adding and removing elements to the start or the end of a list).

In this course, we will survey several fundamental data structures including arrays, ArrayLists, Linked Lists, Stacks, Queues, Priority Queues, Binary Search Trees, Heaps, Hash Tables, and Graphs. We will be particularly interested in algorithmic efficiency, and to this end we will discuss techniques to precisely analyze the number of steps that an algorithm takes. Additionally, we will examine several key algorithms related to these data structures, including graph traversals (BFS, DFS, Dijkstra) and spanning tree algorithms (Prim, Kruskal).

There will also be regular smaller programming problems, as well as programming projects throughout the course, in order for students to gain practice both implementing the data structures and developing a larger-scale code base.

If there is time at the end of the course, we will discuss what it means for two graphs to be *the same* and the consequences surrounding implementing a `.equals()` method that is both efficient and correct. This is the Graph Isomorphism problem. We will focus on the *Color Refinement* or *1-dimensional Weisfeiler–Leman* algorithm, including its power and limitations. The Weisfeiler–Leman algorithm itself is both highly accessible and has deep connections to other areas both of theoretical and practical interest, including computational logics (Descriptive Complexity Theory), optimization (Sherali–Adams Hierarchy), combinatorics, and machine learning (Color Refinement has the same distinguishing power as Graph Neural Networks). Our foci, however, will be on understanding how the algorithm works and to gain practice implementing the Color Refinement procedure, rather than exploring the connections to other areas.

My area of research is in Graph Isomorphism. If you are interested in learning more about this area, including in doing research, please don't hesitate to reach out! I enjoy working with students on research projects and supporting their success!

2.4 Learning Objectives

There are several key learning objectives for this course.

- Students will implement key data structures such as ArrayLists, LinkedLists, Stacks, Queues, Heaps, Hash Tables, Binary Search Trees, and Graphs in Java.
- Students will apply these data structures to solve problems.
- Students will implement non-trivial Java programs.
- Students will analyze the time complexities of algorithms.
- Students will analyze searching and sorting algorithms.
- Students will work through graph algorithms such as tree traversals, breadth-first and depth-first traversals, Dijkstra's algorithm, Prim's algorithm, and Kruskal's algorithm.

2.5 Course Text

I will loosely follow OpenDSA: here. This resource is freely available online.

Remark 1. Many of the algorithms we study have minor variations, which may impact the final answer or intermediary steps. The official version of the algorithms will be those presented in class (and not in supplemental texts). You are responsible for using the version of the algorithm presented in class. **Any deviations from OpenDSA will be clearly documented.**

3 Course Structure and Grading

There will be three components counting towards the course grade: Homework, Quizzes, and Engagement. Homework and quiz questions will be graded according to the following scale: Outstanding (T), Demonstrated Proficiency (P), Progress (PR), Significant Errors or Misconceptions (ATT), No Attempt (NA). Grades of Outstanding and Demonstrated Proficiency count for full credit, while grades of Progress, Significant Errors or Misconceptions, and No Attempt do not count for credit. There will be opportunities to revise and resubmit work on which you did not receive a grade of Outstanding or Demonstrated Proficiency. More details to follow.

3.1 Written Homework

Homework will be assigned regularly, with clearly posted deadlines. You are responsible for being aware of both the **dates** and **times** for these deadlines. Late homework will not be accepted, unless prior arrangements are made or in emergency situations. Please discuss with the instructor as soon as possible if you have a situation that may warrant an extension. Please submit your homework via OAKS.

- There will be a regular written homework. The written homework must be **typed** using LaTeX. Diagrams (e.g., graphs, trees) may be hand-drawn and embedded in the LaTeXdocument as an image and **oriented so that we do not have to rotate our screens to grade your work.** Please note that **handwritten solutions or those prepared without LaTeXwill not be graded.** Similarly, **if we have to rotate our screens to grade your work, then your wil will not be graded.**
- Both your **name** and **student ID** must be included in the appropriate fields. If I am not able to determine who submitted an assignment, then that submission will not be graded.
- The first question on every homework will be an honor code agreement. Failure to indicate that you have upheld the honor code will result in your assignment not being graded.
- You are welcome to discuss the problems with your classmates, as well as reference outside resources. **Anything you submit must be in your own words and reflect your understanding of the material. You should be able to explain your solutions to the instructor, such as in an interview grading session.** If there are any questions about this, it is your responsibility to contact the instructor reasonably ahead of the submission deadline. **Looking up solutions or copying from other sources (including your classmates) is an honor code violation.** You must **cite** any resource (other than the course text, instructor) that you use. This includes any classmates with whom you collaborate. Failure to cite your sources will be treated as an **honor code violation.** See Section 3.7.
- Posting to online forums for help (e.g., Chegg, Reddit, StackExchange, etc.) is an **honor code violation.** See Section 3.7.
- Individual assignments may have additional instructions beyond the syllabus. Students are responsible for adhering to those instructions.

Homework will have multiple problems and may span multiple topics. In order to earn credit for a homework assignment, it is necessary to have earned a Demonstrated Proficiency or Outstanding on all problems. Students may revise and resubmit any homework assignment *for full credit*, subject to the following:

- All revisions are due within 2 weeks after the graded assignment has been returned and no later than Monday, December 4 at 8 PM– whichever comes first.
- You must revise all problems that did not receive Outstanding or Demonstrated Proficiency.
- Any problem that is revised must be accompanied by a reflection. This reflection can take the form of a detailed tutorial on how to solve the problem- this should be written for a CSCI 230 student who is just learning the material. Alternatively, this reflection can be a detailed discussion of your misconceptions when you first attempted the problem and what your understanding is now. **Note:** One-sentence responses of the form "I didn't know then, but I do now.", "I forgot to study.", or similar are **not sufficient**.
- Only one revision is permitted on a given assignment. You may not revise a revision attempt.
- You are welcome and encouraged to discuss problems you don't understand with myself, your peers, and anyone else! The point of the revisions is to help you understand the material. I am here to help you learn the material and succeed!

Fair Warning. Grading first attempts for assignments will in general be my top grading priority, to ensure that I am providing timely feedback on new concepts. As a result, grading for revisions may be considerably slower than for other assignments.

3.2 Programming Assignments

There will be regular programming assignments, which will count as part of the Homework score. These should be completed using **Java 6 or later**. While you are welcome to use whichever editor or IDE you wish, I will compile and run your code via the command line on Windows. Please ensure it is easy for me to do this. **If it takes me more than 5 minutes to configure your source code files to be executed via the command line on Windows, then your assignment will not be graded.**

In addition to submitting your source code, you will also submit a reflection with each assignment. There will be reflection prompts (and possibly additional instructions surrounding these prompts) from which to choose. Your responses to these should be thoughtful and are intended to showcase your learning (please avoid "fluff" and "filler text"). While there are not minimum lengths, it would be concerning if a response to a single prompt was less than two solid paragraphs.

There will also be a prompt where you will document your progress, including discussing any bugs or incomplete components. Programming assignments will undergo two iterations, as follows:

- There will be an initial deadline, where you will submit your current code and reflections. I will then review each submission and provide feedback, including a grade.
- Students who make a good faith attempt at completing both the programming portion and the reflection will be allowed to revise and resubmit their assignment a week later (and no later than Monday, December 4 at 8 PM) for full credit (I expect and hope that everyone will meet this!). The revised source code should be accompanied by written reflections outlining the changes made. These reflections should clearly document the changes you made in sufficient detail as to where another reasonably capable software engineer could follow the changes and work with your code (you may have to do this at a job, with bug fixes/tickets).

If you do not expect to be able to complete the initial attempt by the deadline, please request an extension **at least 48 hours** in advance for non-emergency situations. Turning in a programming project where (in my judgment) negligible progress has been made will not qualify for revisions.

While discussing concepts with others is welcome and encouraged, please be cognizant that sharing or copying code from anyone other than OpenDSA or my lectures is strictly prohibited. All additional code should be written by you, and not copied (which includes retyping) from your peers, online resources, etc. The same Honor Code policies for written homework also apply for programming assignments.

3.3 Quizzes

There will be regular quizzes throughout the semester. In general, quizzes will be announced clearly in advance, though I reserve the right to give pop quizzes. There will be two types of quizzes: (i) in-class quizzes, which are (unless indicated otherwise) closed book and closed note, and (ii) online quizzes, which will be open-book and open-note. All quizzes will be timed and individual effort.

Online quizzes will always be clearly announced in advance. For online quizzes, a LaTeX template will be provided. Students may either type their work or handwrite their work and embed them as images, **provided their work is legible and I do not have to rotate my screen to grade the work**. In either case, use of the LaTeX template is required. Students will have a total of 45 minutes to submit the quiz **as a PDF** to OAKS. The intent is that 30 minutes (scaled for students with disability accommodations) are spent taking the quiz, and 15 minutes are there to prepare the PDF. In practice, students are welcome to allocate the 45 minutes as they see fit. However, as 15 minutes are allocated to prepare the quizzes for submission, late quizzes will not be accepted.

If your internet goes out, you may take a picture (such as with Cam Scanner) and send a **legible** picture (in JPEG, PNG, or PDF formats) within the 45 minute window to the instructor. **I am unable to accept HEIC files.**

As mentioned above, online quizzes are open-book and open-note, but are individual efforts. Consulting anyone who is not a member of the instructional staff about a quiz, which includes your classmates, tutors, and posting online (e.g., Chegg, Reddit, Discord, StackExchange, etc.) constitutes an **honor code violation.** Similarly, all answers must be in your own words and reflect your understanding of the material. Copying from any resource is strictly prohibited. See Section 3.7.

You are welcome to email the instructor with clarification questions, with the understanding that doing so counts against your allotted time and that we may not respond to you in time.

In-person quizzes will be timed at 15 minutes (scaled for students with disabilities). Unless otherwise stated, these are closed-book, closed-note and an individual effort. **Students who are out sick, such as with COVID, should contact me for alternative arrangements. Please do not come to class if you are sick.**

Each quiz will contain at least one problem. In order to earn credit for a problem, it is necessary to have earned a Demonstrated Proficiency or Outstanding.

Your quiz score will be:

(#problems on which you have earned an Outstanding)+
(#problems on which you have earned a Demonstrated Proficiency).

There will be 3 in-class midterms and an in-person final exam, which will be comprised of questions that are similar to quizzes except with more problems. The midterms and final will count towards your quiz score, in that each question on a midterm counts as one quiz question. There will be some replacement as follows:

- For the quizzes taken before MT1, if your quiz score

  (#problems on which you have earned an Outstanding)+
  (#problems on which you have earned a Demonstrated Proficiency).

  is lower than your MT1 score, I will replace that portion of your quiz score with your MT1 score.
- For quizzes taken between MT1 and MT2, I will similarly replace your quiz score with your MT2 score if it is beneficial to you.
- For quizzes taken between MT2 and MT3, I will similarly replace your quiz score with your MT3 score if it is beneficial to you.

So for instance, if you have only demonstrated proficiency (or earned an Outstanding) on 50% of your quizzes before MT1, but then demonstrate proficiency (or earn an Outstanding) on 70% of the questions on MT1, I will– for the purposes of calculating your final grade– record that you have demonstrated proficiency on 70% of the quizzes before MT1. Note that this is an aggregate replacement– I will not be manually updating individual quiz scores.

3.4 Engagement

Most weeks, there will be a reflection on the course content (such as to explain a concept in one's own words or to identify the muddiest point from that week). The exit-tickets will be graded for thoughtful responses, as opposed to correctness. They serve to help the instructor identify what students are (not) understanding.

In order to earn credit for Engagement, students must earn credit for all but two reflections as well as successfully complete the syllabus quiz.

3.5 Cutoffs

Final grades will be issued according to the following cutoffs. Note that you must satisfy **all** of the conditions in each of the columns to receive the given grade. Doing better in one column will not counter-balance lower performance in another.

	Homework	Quizzes	Engagement
A	Full credit on all but two HWs	Proficiency on 90% of quizzes	Credit
A-	Full credit on all but two HWs	Proficiency on 88% of quizzes	Credit
B+	Full credit on all but three HWs	Proficiency on 86% of quizzes	Credit
B	Full credit on all but three HWs	Proficiency on 83% of quizzes	Credit
B-	Full credit on all but three HWs	Proficiency on 80% of quizzes	Credit
C+	Full credit on all but three HWs	Proficiency on 78% of quizzes	Credit
C	Full credit on all but four HWs	Proficiency on 76% of quizzes	No Credit
C-	Full credit on all but four HWs	Proficiency on 70% of quizzes	No Credit
D+	Full credit on all but five HWs	Proficiency on 65% of quizzes	No Credit
D	Full credit on all but five HWs	Proficiency on 60% of quizzes	No Credit
D-	Full credit on all but six HWs	Proficiency on 50% of quizzes	No Credit

Students who do not qualify for a D- will receive an F.

3.6 Regrade Requests

Students have 7 days (including weekends) from when a grade was returned to request a regrade. I am happy to fix mistakes in grading. Other regrade requests will not be considered. When you submit a regrade, please clearly indicate the error made in grading. All regrade requests must be submitted using the following Google form: here.

3.7 Honor Code

I expect students are familiar with policies pertaining to academic integrity, outlined in the Student Handbook. Much of what you will learn about mathematics and theoretical computer science will come from your discussions with your peers. You are welcome and encouraged to discuss the homework problems with each other and with me. It is expected that you work the problems by yourself first, so that you can contribute to the discussion. This policy will be changed, reluctantly, if I find it is being abused. **Your submissions must be written in your own words and reflect your understanding of the material.** Note that you are responsible for citing any resource (including other people) that are not members of the course staff, the course lecture notes, or the lectures. Posting to online forums for help (e.g., Chegg, Reddit, StackExchange, etc.) is an **honor code violation.** If there are any questions regarding this policy, please ask the instructor.

Any acts of suspected academic dishonesty will be reported to the Office of the Dean of Students and addressed through the conduct process. Students found responsible for honor code violations will be subject to the following minimum penalties:

(a) Your grade in the given category will be lowered by 2. So if the relevant assignment is a HW or Programming Assignment, then you would have to demonstrate proficiency on every other HW/Programming Assignment to be eligible for a B+. If the relevant assignment is a quiz, then the assignment will not be graded and the number of quizzes on which you have demonstrated proficiency will be reduced by 2.

(b) You will be reported to the Office of Academic Integrity, which may choose to impose additional penalties.

Honor code violations may result in an XXF for the course, which carries the same weight as an F. The XX modifier denotes that the grade was received for academic integrity violations. **Please do not cheat.** It is not worth it.

4 Course Policies

4.1 Office Hours: Norms and Expectations

There will be a mix of in-person and online office hours (see Section 1.5 for the Zoom link). The purpose of office hours is to supplement lecture and the associated readings. In order to get the most out of office hours, we recommend the following.

- Attend the lectures and read through the lecture notes. In particular, work through the provided examples. These materials are there to help you! **If you are out, such as with an illness, I will be happy to accommodate remote participation.**
- Spend some time working the problems first. Try to identify specific approaches you have made, as well as identify where you are stuck. If you are spending more than 30 minutes on a single problem without making much progress, then I strongly encourage you to seek help in office hours!
- If you wish to discuss specific work, please have it typed up so that you can share your screen on Zoom. It is very hard to help you if your work is on paper and you are holding it up to the camera.
- My goal is to provide hints about homework problems, as well as help students obtain momentum to keep working. In particular, I aim to help students arrive at the solutions on their own. It is completely normal to need time to digest a hint, and then come back to office hours with more questions! Learning CS Theory and Math is an iterative process- we encourage students to iterate!

- Please note that I will neither provide entire solutions in office hours nor grade work ahead of the due date.

Office Hours vs. Email: I am generally happy to discuss course logistics via email (e.g., scheduling appointments, etc.). However, email is usually not a conducive medium for tutoring. If you email me with a question about the homework (and you are certainly welcome to do so), I reserve the right to ask you to come to office hours with your question. Note that this does associate some risk with procrastination, in that you may not get your question answered until after the assignment due date (or after the quiz/exam). Similarly, if you email me late at night, I may not see your email until after the assignment is due. Please plan accordingly.

4.2 Late Work

Late work will **not** be accepted, unless prior arrangements have been made or in case of emergency situations. Extensions can be requested using the following Google form. I recognize that you all will frequently have competing deadlines, including for your other classes as well as personal obligations. There is not always time to meet all of one's deadlines. The way to handle these situations is to communicate reasonably in advance. For non-emergency situations, please request an extension at least 48 hours in advance. In general, I encourage you to ask for what you need. While I will in general try to be flexible for short-term extensions, do note that that requesting an extension does not guarantee that you will receive one.

In the event of an emergency situation which prohibits you from turning in work before the deadline, I may choose to offer alternative flexibility instead of accepting late work.

For long-term emergencies, please talk to me.

Note that missing the homework or quiz deadlines by a couple minutes is not a valid reason for late work to be accepted. Homework due dates and times will be clearly posted, and students will have 15 minutes to submit their quizzes (on top of 30 minutes to take their quizzes). Please plan accordingly.

4.3 Late Enrollments

Students who enroll in the course after the first day of class are subject to the same deadlines as the rest of the class.

4.4 Attendance

Attendance is not required and will only be taken during the first two weeks, for the purpose of attendance verification as required by CofC. Students who have not engaged with class by attending, completing assignments, or emailing me may be reported as having "never attended." If you are sick, please stay home– let me know if this is in the first two weeks, so that you do not get dropped. In particular, if you have COVID, please quarantine until such time as you are not contagious. I will be happy to facilitate remote participation in these instances. In the event that any member of the class (myself included) contracts COVID, I reserve the right to move the entire class online. For in-person assignments, I reserve the right to provide make-ups, utilize (portions of) an exam, or handle the situation in another way that is in my judgment appropriate. Please contact me within 48 hours– or sooner if at all possible– if you anticipate missing an in-person assignments.

Note that ≥ 0 class sessions will be recorded via both voice and video recording. By attending and remaining in this class, the student consents to being recorded. Recorded class sessions are for instructional use only and may not be shared with anyone who is not enrolled in the class.

4.5 Modifications to the Syllabus

The instructor reserves the right to modify any of the policies in the syllabus at any time, particularly as dictated by the interests of learning and fairness. Students will not be graded any harsher than as outlined in Section 3.

4.6 Student Feedback

Student feedback regarding this course is welcome at any time. Those who wish to leave feedback anonymously are welcome to do so using this Google form: here. Students are also welcome to reach out to the instructor via email or in office hours to discuss their concerns.

5 Required Syllabus Statements

5.1 Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, please contact the instructor within the first two weeks to discuss any conflicts with religious events.

5.2 Students with Disabilities

The Center for Disability Services/SNAP is committed to assisting qualified students with disabilities achieve their academic goals by providing reasonable academic accommodations under appropriate circumstances. If you have a disability and anticipate the need for an accommodation in order to participate in this class, please connect with the Center for Disability Services/SNAP. They will assist you in getting the resources you may need to participate fully in this class. You can contact the Center for Disability Services/SNAP office at 843.953.1431 or at snap@cofc.edu. You can find additional information and request academic accommodations at the Center for Disability Services/SNAP website.

If you are not registered with SNAP and believe you may need a disability accommodation, please do not hesitate to contact me.

5.3 Inclement Weather, Pandemic or Substantial Interruption of Instruction

In the event of inclement weather, I will communicate a detailed plan for how class will proceed (if at all). Please prioritize your safety in these situations, including any need to evacuate. If there is a need to evacuate, I will also be prioritizing my own evacuation. The university has allocated make-up days on the weekends to be used if class is canceled for inclement weather. I will communicate in a timely manner for if/how these days will be used.

In the event of a surge in the ongoing COVID pandemic, I reserve the right to make adjustments to the structure of the class. In particular, if there exists at least one member of the class with COVID, I reserve the right to move the course online.

6 Schedule (Tentative)

Class	Date	Topic
1	Aug. 22	Syllabus, ArrayLists
2	Aug. 24	Finish ArrayLists
3	Aug. 29	LinkedLists
4	Aug. 31	More LinkedLists; Stacks, Queues
5	Sept. 5	Binary Search, Mergesort
6	Sept. 7	Quicksort, Hash Tables
7	Sept. 12	More Hash Tables
8	Sept. 14	(In-Class Programming) Hash Tables
9	Sept. 19	Asymptotics (Limit Comparison Test)
10	Sept. 21	Midterm 1
11	Sept. 26	Finish Asymptotics, Start Algorithm Analysis
12	Sept. 28	Finish Algorithm Analysis
13	Oct. 3	Start Binary Search Trees
14	Oct. 5	Finish Binary Search Trees, Tree Traversals
15	Oct. 10	(In-Class Programming) Binary Search Trees
16	Oct. 12	Binary Heaps
17	Oct. 17	Fall Break (No Class)
18	Oct. 19	Midterm 2
19	Oct. 24	Finish Binary Heaps, Priority Queues
20	Oct. 26	Graphs
21	Oct. 31	Graph Traversals: BFS/DFS
22	Nov. 2	Dijkstra's Algorithm
23	Nov. 7	Dijikstra's Algorithm (Complexity Analysis and Proof of Correctness)
24	Nov. 9	Spanning Trees, Safe & Useless Edges
25	Nov. 14	Kruskal's and Prim's Algorithms
26	Nov. 16	Midterm 3
27	Nov. 21	Kruskal's Algorithm (Complexity Analysis and Proof of Optimality)
28	Nov. 28	Review
29	Nov. 30	Review
	Dec. 5	Reading Day
	TBD	Final Exam