
UNH Fall 2023 SEL Survey Instructor Report

Rumi Wheeler (wli5)

Algorithms

CS 758 01, CS 858 01

| Raters | Students |
|----------------|----------|
| Responded | 24 |
| Invited | 42 |
| Response Ratio | 57% |

Course Questions:

Think about the activities in this course that were most valuable to your learning. Describe how these experiences contributed to your learning. Be specific and give concrete examples.

| Comments |
|---|
| The recitations were most valuable. I found that 50 minutes once a week was not enough time. Recitations were often very rushed and we weren't able to discuss everything on the list. |
| Recitation was most helpful for me. It helped tie up loose ends from the previous week and gave a great foundation for the next week. |
| The activities that were most valuable in learning is how to think of problems and innovative ways in thinking to solve a problem. With coming up with different algorithms opened up my mind to the possibilities to solve problems or how to deal with what seem like really hard problems |
| I thought the written assignments were the most helpful, they forced me to use the book more and understand the proofs better. |
| The most valuable thing in this course were the Friday Recitations. Having another way to see problems get worked through outside of lecture, as well as being able to get to know the person grading our assignments was very helpful. |
| I think most of the assignments were good and valuable for my learning |
| I really liked prof Ruml's enthusiasm in every lecture when talking about the topics and found it really engaging. |
| The written homework was the most helpful because it gave us a chance to apply what we learned in class to various problems as well helped to dissect certain algorithms or data structures. For example, one of the first graph homework, we basically did BFS and DFS a couple of times with slight changes, doing so helped us to understand the core components of BFS and DFS and when they're applicable. |
| I think the algorithm construction in recitation, homework 8, and homework 14 were the most valuable to learning. Being able to generate algorithms on the fly is very useful and difficult skill to develop. |
| These experiences contributed to my learning because it helped me be able to understand the algorithms that we have to learn about. In particular, the recitations and classroom lectures because both teachers were straight to the point, and helped explain each topic further to help ensure that we were able to understand the basics of it. |
| We do a lot of examples in class which are all very helpful to understand the course material. Also, though it seems trivial, the song that professor Wheeler sand to us about Turing machines helped me understand the concepts in a fun and memorable way. |
| The most valuable experiences were solving problems, either during recitation, or on the assignments. Working through coding problems and applying the concepts from class was the most useful part of the class. |
| I think we were most helped by the choice of book, it was good at explaining why the algorithms were made like that |
| The homework's were very helpful for retaining the information and studying |
| I feel that many of the examples given in the class were relevant and interesting. Things like sorting, dynamic programming, and NP-Completeness, while challenging at times, are interesting to understand and I feel are important. I liked the examples in class and found things like the drawings on the board helpful. |
| The activities in this class that were most valuable to my learning was how in depth we would go for questions asked in class to improve understanding, and recitations being as well planned as they were. |
| I liked when the professor drew examples on the board but they were sometimes hard to see from the back. He also takes end of lecture questions and dedicates time the next class to answer them. |
| Going to recitation sessions really helped in learning the material in the course. Despite it being an hour long, it mostly cleared up confusion with the material we were learning in class. |
| I felt the recitation periods were very helpful to my understanding, as they often explained core topics in greater detail that I was struggling to understand. The assignments were very difficult, but they covered material that was gone over in class. |
| I think that the activities in this course that were most valuable was the in class examples he did . For instance many of the algorithms have extensive explanations along with them which helped a lot. |
| I found that the weekly assignments allowed me to apply what I have learned. |
| I most enjoyed Assignment 10, where we worked with real OpenStreetMap data. It was fun to calculate my own routes around the state and the New England region. |
| The in class examples and the recitations were incredibly helpful. |

Describe how learning activities in the course might be adapted or modified in a manner that meets your learning needs. Be sure that your suggestions are constructive and specific; general feedback will

not provide your instructor with the information needed to improve the course.

| Comments |
|---|
| The exams were very time restricted. The instructor insured us that there would be plenty of time for the midterm. I found that there wasn't enough time for many students |
| none |
| I think that sticking to deadlines to upload homework is helpful not just for our success but theirs's as well to be able to teach effectively |
| I think the coding assignments were really hard, harder than any class I have ever taken |
| I wish that we were shown more visual examples of the algorithms, as well as walking through pseudo-code step by step to see how it actually works. |
| Assignments 7, 9, and 10 weren't the best and could be redone. Also assignment 7 was changed multiple times after being uploaded and didn't have enough time or help provided to complete it after it was updated the last time and was given an extension and most of the class needed one |
| I think if there wasn't so much dense programming to the point where it doesn't contribute to learning the course content, it could make the coursework easier. But again, it's an algorithms course so it's going to be difficult as a bare minimum. |
| The programs are helpful to understand the algorithm or data structure we're learning about but there are specific things that many people get caught up on that aren't related to the actual algorithm. For example, the testing harness, I often spent a day or two just trying to make the harness work with my code, which I tested it to be correct with the sample input. and many of the issues that made it hard to run my code with the harness were printing or knowing the right command to run the harness (sometimes it differs with the specs). So, an example print in the comments or specs and updating the specs or the —help for the harness would help to shorten that time spent using the testing script. Another thing about the programs is the skeleton code which often lacks comments or explanation of what the structs are. For example, the parsing assignment, there were many structs that were doing a lot of things (nonterminal → nonterminal rules and terminal → nonterminal rules) and the comments helped a little but it was still confusing since there wasn't anything in the skeleton that used those structs, and the mapping wasn't intuitive. So writing more in depth comments/documentation or adding an example that used these structs would help to better understand the skeleton code. |
| I think the course is good the way it is for people like me, whomever they might be. A little more outlining on the skeleton code might be nice, cause it seems some people struggled to do this. |
| The only thing I believe that should be modified would be the lecture slides. The lecture slides were not helpful at some moments due to lack of information on them, and so it would've been more helpful if the slides contained more examples or more information on the algorithm. |
| I think it would be helpful to discuss assignments earlier in the week. For example, Devin answers most of our questions about the homework on Friday when the assignment is due Sunday, so much of the work gets done over the weekend. It would be beneficial to have more class time set aside for work discussing upcoming assignments. |
| I would appreciate talking about the coding assignments earlier in the assignment cycle. We got limited time to talk to Wheeler and Devin about the code until quite late in the week. Wheeler's office hours generally seemed too early in the week, and recitation seemed to be too late. |
| I wish that we did and wrote down more proofs in class. I feel like I didn't understand how to format them or construct them, even after taking math proofs last year. |
| It'd be nice if the homework problems near the end of the semester was more reviewed in the recitation. The recitations were very helpful because we got to discuss past homework, and how to do a type of problem, but it'd be nice if we could just discuss what a problem is asking for. This way I could get the ball rolling on how I'd approach the problem and spend less time on understanding it. |
| I liked the examples in class and found things like the drawings on the board helpful, but it might be helpful to also include some drawings and examples online. I understand there is a certain expectation that coming to class might be more useful, but I would still appreciate something in case there was a situation where class could not be attended. |
| I think that the only suggestion I would make is to make the recitation be in the middle of the week instead of the end. |
| Just the things he drew on the board being hard to see. I appreciated that towards the beginning of the semester they adapted to teach the things for the assignment due on Sunday night the Friday two weeks ahead of time instead of the Friday two days before the assignment was due. I think everything else was good. |
| In terms of the general lectures, they have more often than not been harder to understand. The class is only an hour and a half and runs twice a week, but the issue that arises from this is not having as much time to go over simpler examples to have the students understand how to apply the material we are learning. I think these kinds of examples while going through it with the students would help out tremendously, as students will be more prepared to tackle the assignments, especially in cases where the problem has proofs. |
| I found every weekly assignment to be very difficult. Each one took me nearly all week to finish, and with a new one each week, it was often difficult to shift focus to a new one. As the semester progressed, it became harder and harder to shake off performances I |

| Comments |
|---|
| <p>was not happy with, which cascaded into future weeks. I know this material is not easy, but I feel like even a slight letup of the difficulty would go a long way.</p> |
| <p>I think some of the assignments that would be revised, for instance NP Complete proof was hard to do because we never had an example in class.</p> |
| <p>I think that some of the skeleton codes and assignments needed a little more work. There were quite a few instances when this was either confusing, or needed updating late.</p> |
| <p>I hope the class continues to make the assignments more practical, which makes the content a little more engaging. It seems like the newer assignments (Assignment 7 and 10) are leaning more in this direction. Assignment 9, in its current state, seemed very esoteric when compared to those two.</p> |
| <p>Doing some quick research online, I saw that Minimum Spanning Trees are commonly used to plan the layouts of utility infrastructure, including electricity transmission networks. Flow can also be applied here.</p> |
| <p>I personally like it when assignments in a class are related to each other, like in CS 520 and 720 where we wrote an assembler, linker, and emulator for the VMX20 architecture. It would be nice if we could have some of the assignments be related or connected in this class as well.</p> |
| <p>For instance, A9 could be themed as "The NH Electric Co–Op has commissioned you to plan new lines over the town of Jackson, NH" or something, and you find a minimum spanning tree to plan where that infrastructure should be optimally placed. A11 could be "The NH Electric Co–op has a set of new customers at n locations pulling $[x_1, x_2, x_3, \dots, x_n]$ kWh of power over some unit of time. Can this load be satisfied on the electrical network?" Transmission lines have a capacity they can transmit, and can be modeled as a flow network.</p> |
| <p>Public water or a natural gas pipeline could be a better example for flow. Or traffic flow on roads.</p> |
| <p>Having one assignment where we do heuristic analysis on or are given a special case of an NP–complete problem and implement an approximation algorithm would be nice. It was mentioned, and I heard from a friend that this was in the course material at one point. It should probably be added back.</p> |
| <p>More examples maybe? I'm not sure. I found the way Professor Ruml taught was good already.</p> |

What did you do in this course to promote your own learning?

| Comments |
|---|
| Get started on the assignments early. They may take anywhere from 10–25 hours |
| attended everything and completed everything |
| Read the book |
| I mainly reviewed the lecture slides in my learning. I have never been good at keeping up with textbooks, and so I did not get as much out of it as I should have, |
| Used slides and additional resources |
| I attended all lectures and tried to get extra help in programming. |
| I read the book when the lecture didn't make sense |
| Did all homework and attended every class, Only made it to most recitations however. |
| researching online and watching youtube–videos. |
| I studied frequently as well as trying my best to start the assignments in a timely manner to give myself ample time to complete the programs. |
| I read forums, read the textbook, and asked annoyingly simple questions to Wheeler and Devin. |
| Read the book sections on whatever we were going over. |
| I watched YouTube videos and read the textbook. |
| I gained valuable hands–on experience learning real and relevant topics. Many of the assignments were very time consuming (taking my upwards of 3 hours at times to solve a single problem on the homework) but I still feel the general topics are valuable. |
| In this class I spent about 10–15 hours every week working on programming, as well as using the textbook provided to assist in my gaps of understanding. |
| I studied and did work outside of class. In lecture I took notes on the notes he handed out in class from what he said. |
| I conducted research on my own, but also took some time to understand the slides. Sitting down to understand it helped quite a bit, as I would also try examples out on a separate piece of paper in order to have a sample execution to reference. I also attended most lectures and all recitations in order to get the things I needed to do the work for the class. |
| For this course, I worked on projects as soon as they were assigned, created study guides for the final exam, and regularly scheduled office hours to make sure I was understanding the assignments correctly. Some way or another, I worked on this class every day from the start of the semester to the final exam, in order to keep it fresh in my mind. |
| A lot of learning on YouTube and going to office hours. |
| I worked for a long time on my assignments, and studied the course material. |
| I read through some of the chapters in CLRS if I had difficulty understanding content in the lectures. To really digest some of the homework problems, I had to take breaks and sometimes take naps. |
| Not enough! I took avid notes but I wish I had spent more time outside of class working on this. |

Using the following scale, please rate your recommendation of this course to other students whose academic preparation and program are like yours:

| Not Recommended | Neither Recommend Nor Discourage | Recommend | Recommend Strongly | Resp |
|-----------------|----------------------------------|-----------|--------------------|------|
| 4% | 21% | 46% | 29% | 24 |

Would you recommend this course to other students whose academic preparation and program are similar to yours? Why or why not?

| Comments |
|---|
| It had value and I learned a lot. However we sacrificed a lot of time for the course was more than the 12 hours per week. If there was enough time to do the assignments, I feel we could have done way better |
| It is required. I had heard that Prof. Ruml's version would be more difficult, but it is plenty doable. |
| Yes since it helps to think better about making efficient algorithms, because in data science, those algorithms can take long, and it is important to know how to make proper code so implementing machine learning algorithms is both efficient as well as optimal. However, I do think the learning curve is very high how it is currently set up, so having more set up to get to this class would help to lower that curve. |
| Well we have to take it, but I did learn a lot |
| It is a required course, but other than that, this class has a level of enthusiasm that a lot of other classes lack which makes it much easier to learn. |
| I would but would make sure to say its a lot of work and alot of time needs to be done |
| For someone who has the same academic preparation as me or similar, if this wasn't a required course to take, I wouldn't recommend it because of how programming heavy it is and how it's assumed that you already have this knowledge. Otherwise it's a great course and help is accessible. |
| This course was required so I didn't have much say. But the course is very intensive but interesting, so if other majors are looking for a challenge, this course would certainly be a challenge |
| Yes. It is required and gives you the ability to prove your own programs correctness. |
| Yes, because it helped me understand the fundamental of algorithms, as well as why they are so useful and needed in our day-to-day life. It also was helpful for my technical interviews, because they would mention the basic of an algorithm, and I would be able to explain it further since it was covered in class. |
| I would recommend it. This is the first class that I thought the way I was coding really mattered. Previously, the efficiency of my algorithms or methods I used were not as important, but this class really shows us the difference in what type of algorithms we use and why some are perfect for some situations, and terrible for others. |
| I would discourage students with a similar lack of programming background from taking this course. Without the requisite basics of programming this course is extremely difficult. I'd recommend someone with a similar background to me to be prepared to sacrifice a semester of work/life balance. |
| Yes, it's difficult but teaches a lot of important concepts. |
| Yes. This class will 100% force you to write better code because when you know you can do better, you will do better. |
| I would recommend this course to to others, especially just for the topics covered. I think some of the ways things were delivered are not perfect (no late days, lack of a canvas page) and the material is very difficult, but it is good information, and it is valuable to learn. |
| I would recommend it. While this class was very challenging, it was essential to build good skills for a computer scientist. |
| I would recommend taking this class since a lot of the questions we do in class have shown up as very similar interview questions. |
| For those that eventually come into this course, this is the hardest course of the entire CS curriculum in my opinion. Coming into this with only about a year of C/C++ experience through UNH courses was difficult. The proofs can also be tricky to understand and thus you may need to ask a lot of questions. Personally, I wouldn't recommend it, but I can see some areas of the course can be useful later on. Overall, it doesn't hurt to try, but the best thing is to be prepared for it, as a lot of time outside the course is going to be spent doing work on the course, and it's especially hard when taking more than 4 classes. |
| This class is required, so I cannot say my recommendation matters much. However, I personally found this class to be the single most difficult, work-intensive and stressful one of my entire educational career, so I would mention that to anyone who is about to take it. |
| Yes, because it is a very interesting class and definitely made me a better programmer |
| I would, I think that the concepts taught are very valuable. |
| This course, the way Wheeler Ruml teaches it, is essential for senior CS students. It's challenging and it very much changes the way you think about problems. |
| Sure. It's really important information. I'd suggest other people bring more C programming knowledge to the class than I have. |

About the instructor:

Using the following scale, please rate your recommendation of this instructor to other students:

| Not Recommend | Neither Recommend Nor Discourage | Recommend | Recommend Strongly | Resp |
|---------------|----------------------------------|-----------|--------------------|------|
| 0% | 17% | 46% | 38% | 24 |

Would you recommend this instructor to other students? Why or why not? Please be specific and constructive.

| Comments |
|---|
| Yes, since. They are energetic about the material. The slides get a little rough in the middle of the semester. |
| Yes. very intelligent. |
| When it comes to an instructor that is passionate about what he teaches, there is no one who is more than Professor Wheeler. He really tries to make you think critically and he can be helpful to give you the knowledge to succeed. I do feel like however his assignments can be tough and is too stuck to his schedule at times to always give the students a chance to get the best chance to succeed. He does show to want to make improvements which I can appreciate. But overall I would recommend him to have someone who really wants to teach you. |
| Prof Ruml is a tough teacher but he is really enthusiastic and makes class very fun. It is a push-and-pull relationship because the class is really hard and you have to buckle down, but you do learn a lot. |
| I really enjoyed learning from this instructor, the enthusiasm and general explanations made this course much easier to take. |
| He is really good an capturing attention and lecturing, but felt like the midterm was a mess with having some people start while others had to wait for their exams printed. Gave an extension for an assignment over thanksgiving but didn't give one for an assignment before the midterm which felt like it was more warranted |
| I think he's an excellent professor when it comes to teaching in computer science and his enthusiasm for what he's teaching is always a pleasure to see when coming to class. He's very thorough in explanations and shows interest in students' questions and ideas about the course content. |
| Prof. Ruml is very enthusiastic in class and it makes learning the material entertaining. However, he goes very fast through the material and I often need to revisit the book or lectures to fully understand the topic. This fast-paced-ness can also be felt when students ask questions and he rushes through them or gets visibly annoyed at times, and it can be discouraging to ask more questions. |
| Yes. Professor Ruml is very enthusiastic and knowledgeable about his class and is very knowledgeable. His lectures are enjoyable to listen to. |
| Yes, because the instructor was able to answer most of the student's questions, and would try to understand what was being asked to which he would try his best towards responding/answering it. It was important when the slides weren't that helpful, and so him being able to answer what was being taught made the course easier to get through. |
| I would recommend Wheeler. Though he is a tough grader and difficult professor, he is clearly very interested in what he is teaching which makes the classroom environment more engaging. I do think that he sometimes grades too harshly, this can be seen by the very low test averages each year. |
| Probably. Wheeler has very high expectations for his students. Part of this, I think, comes from the fact that he optimistically believes everyone is as sharp as he is. This is not always true. |
| He makes class fun and engaging, with interesting metaphors and scenarios. |
| He was vey informative |
| Professor Ruml has many good qualities, but there are a few which I would warn of before taking one of his classes. Starting with the good, he is immensely knowledgeable. The first day of class he pulled out all kinds of books on algorithms that he has read and recommended we should read that he personally has read and his passion for the topics shows. He comes to class up-beat and with a good attitude and is open to questions to help students who are struggling. Often there are times when a student has a question outside of the material we learned and he is able to just think up the solution in front of everyone which I feel is a testament to his knowledge. I think the main thing I would warn students about is he can be slightly unforgiving at times. He knows the information very well and, as such, expects very correct answers. Because of this strive for the best, there can be things that make things harder (like the lack of late days on assignments) which I don't feel support learning on the assignment. I understand the thought that there is value in the struggle and I agree with that, but some of the assignments are extremely hard, and he has acknowledged that they are very difficult, but there still are no late days and grades can be brutal at times if your code is even slightly off. He does curve the letter grades quite a bit which is helpful, and I think makes it more fair, but the grading strategy can still be very brutal. |

| Comments |
|--|
| <p>In summary, I think he is a great professor and a very good teacher, but he also has high expectation for his students and the material in the class is very difficult.</p> |
| <p>I thought he was exactly what this course needed, gotta love devin.</p> |
| <p>He is open to questions and change, and has been changing the assignments throughout the semester so they are to the level of what he taught. I didn't like that he wouldn't change a deadline after he said he would if he couldn't get an assignment out by a certain time. When this happened it was difficult to reorganize my week, where if the assignment deadline has been pushed I wouldn't have had to change anything.</p> |
| <p>An enthusiastic teacher is always nice to have in order to lighten up the mood when entering class. However, when teaching, he doesn't really explain stuff as well as I hoped. I wish he would have taken more time to explain the material. The end-of-lecture questions are always nice to have, but for me it takes a lot longer to absorb the material and fully understand it. I'd still recommend him regardless.</p> <p>As a side note, I'm not sure if this is an issue I have, but I don't like the use of analogies in some of the material, such as minimal phone-tapping and robbing a house. Although they make sense for a specific algorithm and help a lot when going through how it's done, it's a large discouraging factor for me.</p> |
| <p>Professor Ruml is unlike any professor I have had up to this point. In lecture, his passion for the material is evident, and he does a great job with keeping students engaged and the mood light. However, his lecture slides are vague without his descriptions, and can be difficult to study from as a result. In addition, whenever I went to office hours, he seemed overly cautious of giving away the answer to a problem, to a point where the advice that was given was confusing to understand. This in itself is not bad, but for a course as difficult as this one, there is a need for constructive feedback, which was complicated to receive because of this. He was supportive during this time, however, so it was not entirely negative. Overall, there is a lot to like, but also a lot to have mixed emotions about.</p> |
| <p>Yes, he is very helpful and knows what he is talking about.</p> |
| <p>Ruml is very passionate about the subjects being taught. He also teaches the class in a light and happy way, which helps engagement.</p> |
| <p>Wheeler Ruml is a good professor, and his lectures are engaging. I was very rarely bored in his class. His lectures have a mixture of humor, course content, and enthusiasm.</p> |
| <p>Professor Ruml is very charismatic and rather good at conveying knowledge to us in memorable ways.</p> |

Instructor Personalized Questions

About how many hours per week did you spend on this course on average over the entire semester, including lecture, recitation, and homework?

| Comments |
|--|
| 25 hours per week |
| 9 |
| Upwards of 30 hours but around 18–20 on average |
| probably 20 to 25 |
| I spent roughly 15–20 hours a week in this class depending on the assignment type with coding assignments taking much longer. |
| 15 hrs |
| On average, I'd say I'd spend about 4–5 hours on each assignment, and for courses+lectures it would be 3.5 hours total. |
| 15 hours |
| 4 hours for Lecture and Recitation. About 6ish hours for homework, but it ranges widely. |
| 15–20 |
| I spent about 12–16 hours per week on this course depending on which assignment we were given that week. Some took much longer to complete and some took less time. |
| Probably 30+ on average. |
| 12–20 |
| 8 |
| 20 |
| I easily spent 12+ hours on the homework alone every week. Often times I would try to start early but the homework would still consume my entire weekend up until the deadline which made them daunting and made me less motivated to do them. As I mentioned in some other parts, there were some problems that would take me over 3 hours on their own just trying to think through them and understand them before I could create an answer. The homeworks are definitely extremely challenging. On top of this there was also of course two classes a week which are an hour and 20 minutes, and a recitation on Fridays which is only 50 minutes. |
| 17 hours per week. |
| 20 |
| 4 hours every week for lectures and recitations, sometimes less depending on lack of sleep. On homework, anywhere from 12–20 hours outside of class. |
| This class easily took at least 20 hours each week to work on. For certain assignments, this number increased even more. |
| I spent a lot about 12–20 hours per week. |
| Somewhere between 10 and 15 depending on the week. |
| Between 6 to 16 hours outside of lecture and recitation. Around 10 – 20 hours including the classroom time. |
| 13 ish |

What is one specific thing that the instructor should continue doing, stop doing, or start doing, in order to maximize student learning?

| Comments |
|---|
| I feel as though the material was rushed. Because of these the slides should have more detail since we often need to look back at them |
| EOLQ are good to have |
| Stop sticking to the rigid schedule. Continue to be passionate and avid teacher |
| I don't know. The structure of this class does require a lot of assignments but 1 a week is really hard. |
| I would like to see more visual aids when exploring these algorithms, as well as including variable keys in the pseudo-code that is provided in the slides. Looking back at slides that lack understandable variable names makes it very difficult to figure out what the algorithm does. |

| Comments |
|--|
| I think the recitation is good but the assignments seem a little too much for a few and too little for others |
| I think if the programming portion of the assignments were reviewed (in case it hadn't been worked on in years) before being put out for the students to work on (like the skeleton code), it could save time of students having to struggle with bugs that shouldn't have existed when compiling. |
| The instructor continue to give programs and assignments, although with some tweaks; stop going so fast-paced and adhering to the schedule so strictly; and start adding a drop for the assignments or a break from assignments and exams somewhere so we can rest |
| I don't know. |
| He should continue the end-of-lecture questions, but should try adding more little bit more helpful information to understand the lecture slides. |
| He should continue to give lectures with the same energy, as it promotes an engaging environment and makes paying attention easier. |
| I think that programming assignments should come after the word assignments. |
| Continue having the same lecture style Stop giving broken skeleton code or assignment docs that have commands that don't work Start providing full proof example in slides for reference |
| I really feel that the lack of late days and the deadline being at 11:30 are unnecessary. Of course if you had late days with a penalty it would still incentivize students to get it done by the deadline, but the lack thereof just makes it harder for the student who is likely struggling with the material. I understand that there are homeworks every week and you wouldn't want a student to fall too far behind in the class, but the alternative is that student working hard on the assignment, struggling, not fully completing it by the deadline, and then on top of the struggle and shame they likely feel for not finishing by the deadline the further penalty of grading their incomplete/poor work harshly can be very hard on a student with other classes that have demands. I feel that if you are going to limit students to getting it done by the deadline and no later you have to compromise by being slightly easier on the grading and giving partial credit. |
| Stop giving so many assignments, give some breaks to solidify knowledge learned. |
| I've mentioned this already, but the best thing the instructor should start doing is providing simpler examples and going through them with the class to keep the class engaged in the learning. The instructor should also keep being enthusiastic, as it helps students become more engaged |
| It sounds cliché, and is not always the best advice, but simply letting up on the difficulty just a little would do wonders for this class. The material in itself is as complicated as other CS classes, and there are questions that exist to match, but when the assignments are of this difficulty and due every week, it is easy to get lost, lose confidence, and struggle with burnout. The very first set of slides cautioned against impostor syndrome; by the end of this semester, I can say I felt it multiple times as I worked on these assignments. |
| Keep up the passion about for the algorithms , it made it very helpful to be in class. |
| I think that sometimes things can be oversimplified, such as proofs. This isn't always the case, but maybe spending more time on things like structure could be helpful. |
| Asst 7 needs the most work. |
| I wish we were allowed to use other system-level programming languages. NOT JS or Python, or Java. Languages like C, C++, or Rust. |
| It was brought up that these languages have "temptations" in their standard libraries, which I agree with... but so does C to an extent. Just say, if you use these languages, you can only use such and such things in the standard library. For Rust, it's easy to check the "use" statements to see what features students are using. |
| Dealing with memory errors isn't the focus of the implementation assignments in this class... implementing algorithms is. It would be nice to have the option to have more modern languages that put an emphasis on memory safety and avoiding undefined behavior. |
| The way you ask for our feedback and patiently wait to answer our rambling questions is good. You ask clarification when you don't understand what we want. Ideally I'd like a little bit more flexibility concerning turning in late work. |