COURSE REPORT

Course	CS2040S - DATA STRUCTURES AND ALGORITHMS
Academic Year/Sem	2025/2026 - Sem 2
Department	COMPUTER SCIENCE
Faculty	SCHOOL OF COMPUTING

Note: Class Size = Invited; Response Size = Responded; Response Rate = Response Ratio

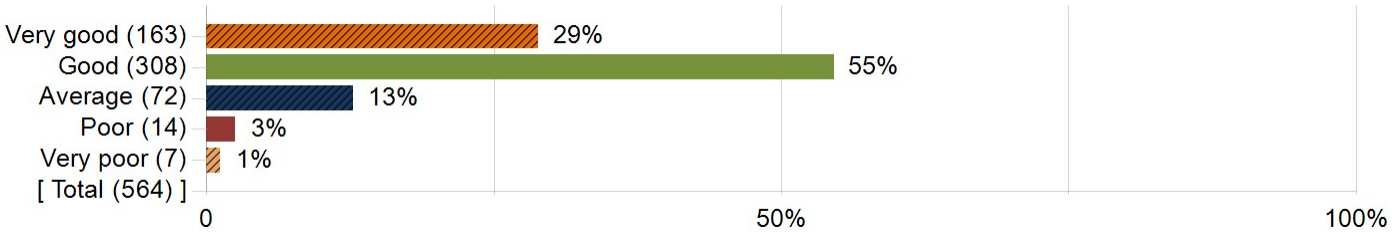
Raters	Student
Responded	567
Invited	711
Response Ratio	80%

Instructors of large courses (300+ students) can now benefit from an AI-powered tool developed by ODI in collaboration with PVO. This tool analyzes qualitative student feedback to provide quantitative summaries, offering valuable insights alongside the traditional reports. Access the summary here | details about the tool here. For inquiries or suggestions on improvement, please contact Ms ONG Mui Hong (Director TEL) at muihong@nus.edu.sg

1. Overall opinion of the course

Distribution of Responses

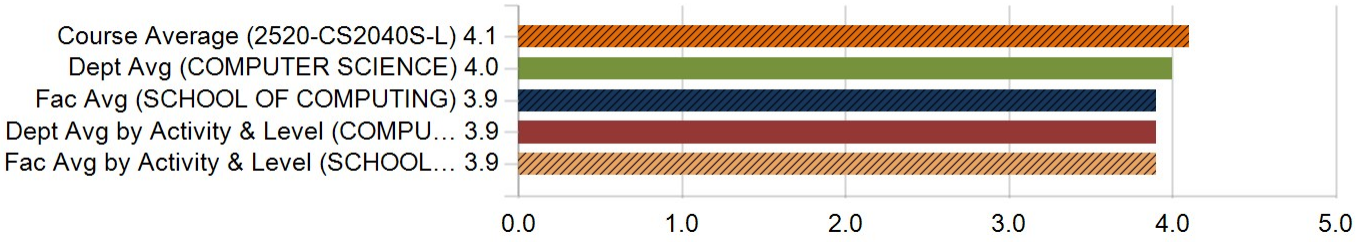
What is your overall opinion of the course?



Statistics	Value
Response Count	564
Mean	4.1
Median	4.0
Mode	4
Standard Deviation	0.8
Positive Feedback	84%

Rating Scores

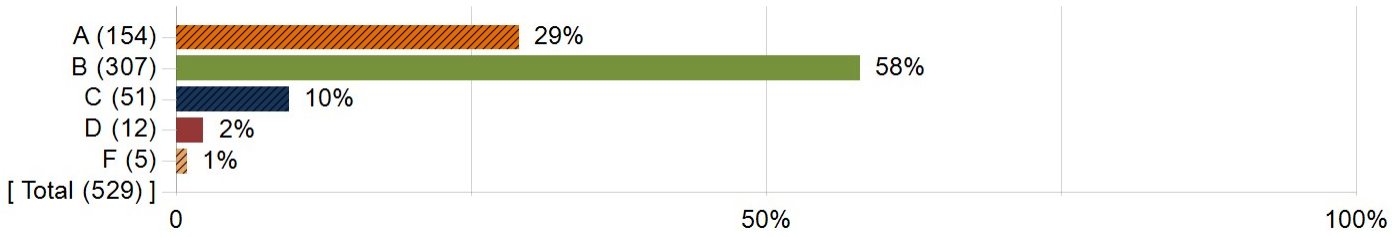
Question	Course Average (2520-CS2040S-L)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
What is your overall opinion of the course?	4.1	0.8	4.0	0.9	3.9	0.9	3.9	0.9	3.9	0.9



2. Expected Grade

Distribution of Responses

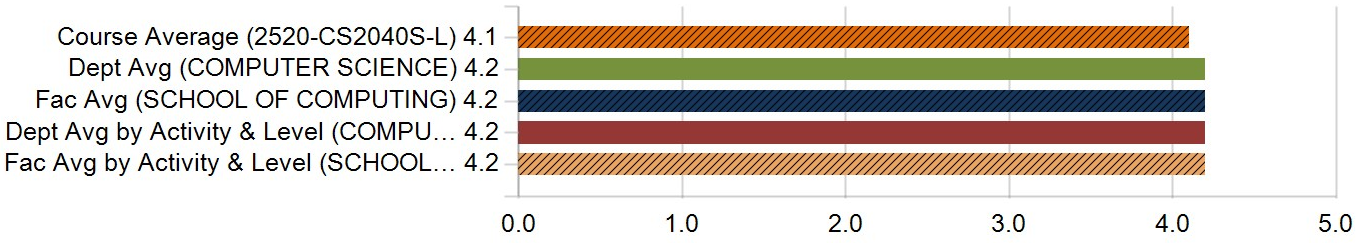
The grade that I am most likely to get in the course is:



Statistics	Value
Response Count	529
Mean	4.1
Standard Deviation	0.7

Rating Scores

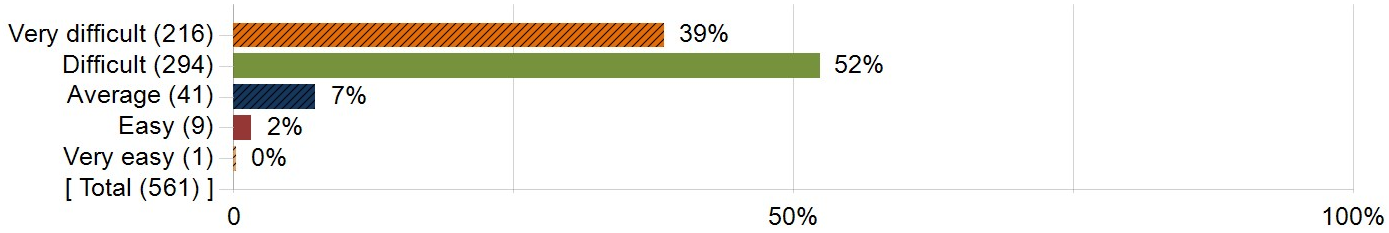
Question	Course Average (2520-CS2040S-L)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
The grade that I am most likely to get in the course is:	4.1	0.7	4.2	0.7	4.2	0.7	4.2	0.7	4.2	0.7



3. Difficulty Level of the course

Distribution of Responses

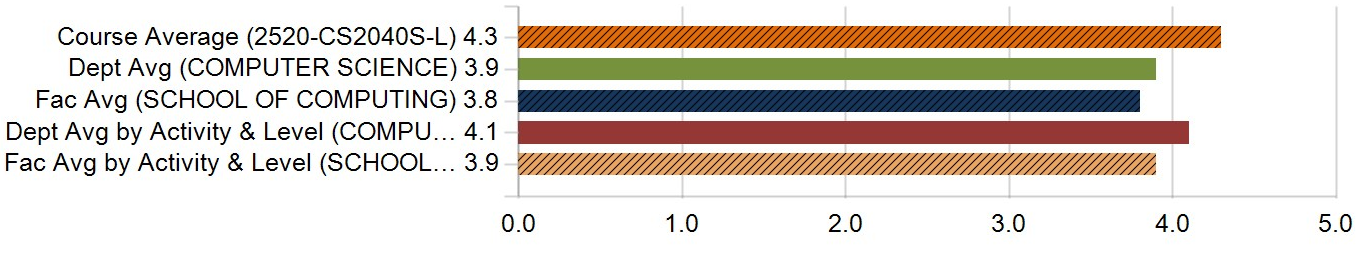
I rate this course as:



Statistics	Value
Response Count	561
Mean	4.3
Standard Deviation	0.7

Rating Scores

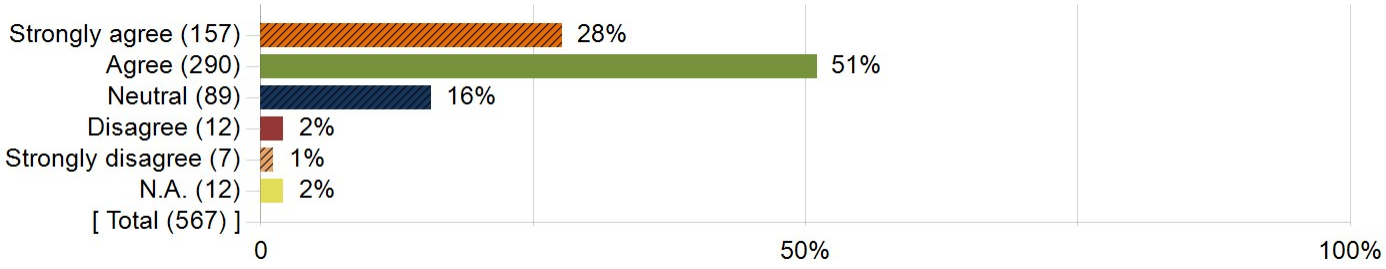
Question	Course Average (2520-CS2040S-L)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
I rate this course as:	4.3	0.7	3.9	0.8	3.8	0.8	4.1	0.7	3.9	0.8



4. Technology integration

Distribution of Responses

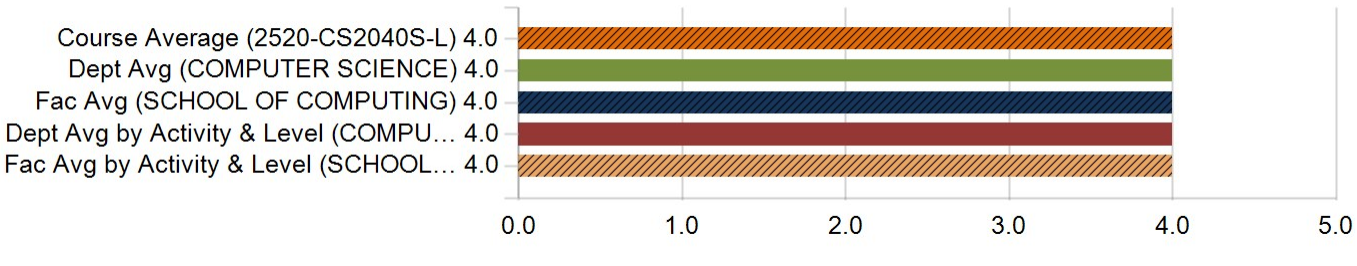
The integration of technology—blended learning, digital tools, AI, and online resources—has enhanced my learning experience.



Statistics	Value
Response Count	567
Mean	4.0
Standard Deviation	0.8

Rating Scores

Question	Course Average (2520-CS2040S-L)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
The integration of technology—blended learning, digital tools, AI, and online resources—has enhanced my learning experience.	4.0	0.8	4.0	0.8	4.0	0.8	4.0	0.8	4.0	0.8



WHAT I LIKE / DISLIKE ABOUT THE COURSE

What I liked about the course

Comments
Interesting concepts
learnt about algorithms, learnt problem solving skills, thinking out of the box, using tools we learnt to solve new problems
all course related information and work available on coursemology

Comments
This course requires us to apply the different data structures and algorithms, and probes our thinking during the process. The questions are all quite related to our everyday life, and it shows me how the knowledge we learn can be applied to solve everyday problems.
1) Cutting edge material 2) Very good problem sets, assignments.. 3) Interesting exams 4) Enthusiastic profs
I think the course covers an appropriate amount of content, and teaches them decently well.
The content is overall quite interesting, aside from the part that I may not receive a good grade in the final haha.
Problem sets are good practice but not really applicable and relatable to tutorial
I liked that the slides provided are very comprehensive and serve as good notes
The course is well structured. The exam test topics that are according to what is taught in lessons.
The algorithm isvery interesting
engaging lectures from eldon
The course teaches Data Structures and Algorithms, which are important for technical interviews and getting jobs. The course also teaches how to solve problems efficiently.
fun
almost nothing
None
Forces me to think and have good mental models and have good problem solving techniques
na
Great encouragement of critical thinking and problem solving while going into depth about the basics of DSA. I really liked the range of problems we got to solve and the wide range of topics covered, including those that hint about further topics relating to algorithms.
.
The problems were fun and I really felt that I learnt a lot.
algorithms are interesting and useful
Good choice of topics
engagig lectures
i like learning about data structures and algorithm so i think its a really good course
i love data structures and algorithms :>
I liked the lectures.
Very useful for job interviews.
Interesting fun course, but can be difficult
I like how this course focuses on our communication of our high level ideas while still achieving a balance with our actual implementation of our solutions, with the focus of preparing us adequately for our technical interviews. I think this is a good direction for the course as it grounds this course on the practical side of things rather than the more theoretical side that most university courses in NUS tend to be, making this course highly relevant for us even outside of the university academics.
Good use of technology to teach the course. The slides are also always interesting.
Coursemology assignments were very good
teaches algorithmic thinking well
Having a problem sets where you actually have to implement the problems.
nil
The course content is very useful in this field and tools such as Coursemology encourage a hands–on approach that promoted understanding of the concepts.
pseudocode
i like that i can write in pseudocode and there is a new ai helper in coursemology.
Some of the concepts were very interesting to learn
Course probed my thinking

Comments
algorithm that we learnt were very interesting
I love the topics covered in this course
I really liked the content and how it has been taught. I think the course gets us to think a lot for ourselves and the things we learn seems both useful and interesting.
I liked that this course has given us a broad sense of data structures and knowing when to use it.
nil
eldon was a good prof!
Gamification through Coursemology Videos published help me catch up if i miss lectures Tutorials to go through problem solving
Exercises, although tedious
nil
It is helpful and relevant to me
Interesting content, quite fun
Content was interesting and relevant Well–structured and easy to follow Good balance of theory and practice Assignments helped reinforce concepts Lectures were clear and engaging Useful for real–world applications
Enhanced critical thinking
problem sets were quite interesting
The course topics are interesting and practical for job interviews.
it was challenging!
More engaging than CS2030S
Prepares me for interviews
Brain teaser
fun problem sets, i like the interactive format on coursemology
Good intro to DSA
Useful and practical content learnt
N.A.
Its relevance and the problem solving aspect!
interesting concepts i havent touched on before
Educated me well enough to attempt most of LeetCode questions and enjoy DSA
The course had a balanced workload and is taught very well.
Nil
Tutorials are extremely active and cover content including and beyond the lectures
I like the anime memes on the slides.
–
Gave good introduction to DSA.
I am able to learn new algorithm and able to see the application of these algorithms
teaches very useful stuff
I love data structures, sorting algorithms, and pseudocode, this course has them all! I like the optional practices.
The course guides us to consider problems from a broader view instead of struggling with minor things during implementation. This helped me to establish a greater view of the industry and the major.
Pseudocode in exams. Using IntelliJ instead of Vim. Coursemology xp.
Allowed me to think more deeply about data structures and how to apply these data structures to better solve problems that I might have not seen before.
prof eldon is the goat. Also the coursemology problem sets are very useful

Comments
coursemology
eldon
Recorded lectures; I have speech recognition issues that are aided by them. Problem sets. I love problem sets.
I liked that materials are fairly well–organised and many different sessions (tutorials, lectures, recitations and missions) to ensure that concepts are internalised.
Complex
Enhance our critical thinking ability and problem solving skills through introducing various data structures and algorithm. Makes us appreciate its application in real life,
the problem thinking and fun logic
Scoreboard system Puzzles to solve in coursemology Emphasis on knowing how to do stuff rather than making a surgical precision implementation (especially considering no class really teaches the Java language explicitly) Engaging classes A final fist–pumping competition in the last week of lectures to help us review what we learnt LETS GO BOYS WE STAY WINNIN' I just really like everything about the course man, algorithm is my jam
The concepts are pretty interesting.
Very interesting and highly useful.
Found a passion for algorithms here
coursemology was a good platform that was stable and also smooth in handling the entire mod
Do things fast, do things correct, learn better way to do thing :D
dynamic programming
Yes
its interesting
it is fun to solve the problems
problem sets are interesting
Eldon motivated me everything about this course
The course has taught me useful DSA skills and knowledge that I will definitely use in my further studies and later in the workforce.
nil
Helped me do LeetCode.
I like the fact that I did not need to implement the solutions in Java fully, and that during the exams, the high–level idea/concept was good enough. This would have been EXTRA tough to do, especially when understanding how to apply a certain data structure or algorithm to a problem at first.
–
coursemology was fun
i liked the face to face lessons with the tutors
This course is a critical course in the degree and is well taught by a knowledgeable course coordinator with well–respected supporting staff.
content
I liked solving the problems.
NIL
sa
eldon’s lectures
I like the recitation problems.
I like the concepts, like its content heavy but solving does seem interesting
This course emphasis the importance of problem solving and not blindly typing the code.
The lecturer
helped me to learn about algorithms and data structures which i can now use

Comments
Interesting to learn about various data structures
It's a tough course, almost like an IQ test every day, but I find it fun.
eldon
The course content is very intellectually stimulating, it's fun to do problem sets, but can be quite difficult at times
–
The problem solving involved and how it's actually quite fun to think of unique solutions to the problems. I also like the usage of coursemology to make learning more fun :)
i dont know
The focus on problem solving rather than pure regurgitation
nil
the problem solving exercise and the content i learned from it
Intersting
This course served as a good intro to data structures and algo and has allowed me to see how useful these things are in solving problems in general
Activity on Coursemology was nice, can easily get help
I loved actually learning problem solving skills, though applying it well is extremely difficult and should be.
the teaching was good
The content taught
Learning about algorithms
NA
i like eldon
Fun and interesting
require actually thinking lol, not just textbook learning like 2030s
i like prof eldon
fun to learn algorithms
Engaging and interesting
Course content is structured well, with every week building on what was taught previously
Interesting problem sets Interesting content
Problem set scenarios
One of the first CS modules that really gets you started with the different data structures that can be applied outside (though definitely not the same and will certainly need many modification, but I stand by the fact that it teaches you enough that you have a decent foundation to learn more outside)
The course can be engaging, it teaches students to use critical thinking.
Teaches algorithmic thinking and analysis. Even though it is extremely challenging and difficult
Useful for computer science and good pace
interestingg
The focus on interview preparation and high level ideas
Very fun to listen to the lectures and recitation. It's a very good learning experience and I feel like I've learned a lot.
Centralized resource (coursemology) helps with keeping on time with assignments
problem solving experience
I really enjoyed the course content, and the materials provided by the teaching team. I also especially enjoyed Prof Eldon’s teaching style, and how he goes through sample problems and guides us towards the solution step by step instead of just showing the solution and proving it works.
Application and nicely theoretical
concepts very relevant and necessary
Interesting and Engaging
I learned a lot useful ideas.

Comments
tutorial questions was interesting to see the application of DSA
nil
Good lecturer
Its quite interesting and it makes me think
Cool concepts, a lot abt graphs...
online lectures
It's a lot of mathematics
It hones our thinking of programming
Covers a wide range of important concepts
Problem solving is fun :)
I think this course is very helpful for our technical interviews.
notthing
The content is interesting and the lecturer is engaging
I love the way the material is tested. Instead of concise accurate answers that differentiate based on accuracy or preciseness, it rewards deep understanding and creative thinking even in logical thinking
na
The class is interesting and also challenging. I learned a lot from this conversation
taught me good basics of data structures and algorithms
I loved learning about algorithms and applying them to various problem solving questions. I truly enjoyed the process
The content itself is interesting and likely useful
Coursemology use was good, lectures are good and tutorials are fun. Solutions can be handwavy
Thought–provoking.
I like the problem sets, the tutorials, eldon's lectures
Problem solving
Does a great job at providing lots of engaging content for students to practice and and self–learn, a great introduction to data structures and algorithms overall.
The use of problem sets, tutorials and recitations to promote active, hands–on learning which helps in the learning and application of the concepts and algorithms covered. Supplementary materials such as practice problems ease students to further explore each topic to gain exposure and grasp of the content involved. The content taught and the problem set questions are enjoyable in a way that emphasises their practicality and relevance in real–life applications, providing motivation and an inducive learning environment. If they were to be emphasised more in grading, students would be rewarded more for specifically understanding the concepts and algorithms than being exceptional test–takers.
useful
this course is more about problem solving as compared to blindly memorising the materials, which is fun to me
i mean every CS major must know DSA so it's quite important and relevant. it's quite fun cos of coursemology and the problem sets are fun and make me think
the lectures were well prepared and well done
Gives us the introduction to different data structures we will use in the future
Interesting algorithm helps us build a more efficient system
How useful and applicable it is in interviews. I also like how this course trains us in thinking using different algorithms.
good exposure ti DSA
– Interesting course which teaches us important fundamentals for computer science – Fun lectures by Prof Eldon and interesting tutorial sheets – Interesting problem sets which were quite fun to solve
the content is very interesting
This course taught me a lot of new things related to Data Structure & Algorithm
interesting content
It is mentally engaging, it challenges me to solve interview–style questions
Engaging lectures

Comments
I like algorithms
The concepts taught are really important and useful for my career.
Interesting content.
no code writing required
very interesting course with many high level concepts and ds and a taught which was new to me
–
challenging
The concepts are interesting and important.
nil
Eldon and his Uniqlo pikachu tshirt. Also how much less (discrete) math there is compared to previous years.
fun
Great lectures
It was very interesting that we get to get solve different questions
The profs.
Foundation to data structures
This course isn't my strongest suit, but the lecturers, TA and recitation tutors are all massively helpful in my journey.
Eldon Chung.
The content and the professor, as well as the materials given in tutorials and recitations
It is somewhat interesting and I am kind of interesting in solving similar types of problems in the future, although this interest is kinda dampened by the difficulty (of exams). I liked that the lectures are not compulsory to attend and are recorded, as I will not be able to follow the explanations if I had attended physically.
Actually very applicable for CS students –> Definitely helps in OAs
I like Prof Eldon. I like that exercises. I like the recorded lectures.
It is intellectually stimulating. There's no rote memorisation but a lot of problem solving. It prompts me to actually apply my knowledge rather than just understanding the theory
Knowledge
almost nothing actually
Very interesting and teaches you to think about computer science in a different way.
more focus on problem solving and less on the intricacies or syntaxes of specific coding languages
Its a challenging course that requires you to think a lot about how to solve a problem
good at explaining basics of DSA
I like the way it teaches us problem solving skills and how frameworks for approaching questions are explained.
Taught me the beauty of dissecting code and solve problems intuitively
nil
i like the slides
Problem sets and Coursemology

What I disliked about the course

Comments
Difficulty is high
Use of Java, Python is more suited
didnt know the weightage of each assessment
I feel like some of the content can be a bit challenging.
1) Extremely high breadth of syllabus: very difficult to follow along with all content in a week. I personally found myself drowning in just understanding the material, and simply not having the time to go deeper and solving a lot of problems. Many times throughout the semester, I found myself spending days understanding and keeping up with all concepts taught that week, and I personally found that I was not able to "truly" practice each chapter because of this volume of content. I do feel that this course rewards people

Comments
with prior experience heavily, however for people starting from scratch it is very difficult keeping up with the material .
Workload is way too high. Lectures, tutorials and recitation already take up to 6 hours each week. Time taken to prepare for tutorials and recitation easily take up another 4 hours. On top of that, there are still problem sets and lecture review quizzes. In total, the mod takes up almost 15 hours per week. It also does not help that course admin is not very organised. Recitations are often released very late, only 1 day before the slot. Sometimes, lecture review quizzes are simply "forgotten to be released". The use of coursemology compounds these problems as there are very often bugs in coursemology and for some of the problems, the grader is almost non–deterministic (submitting the same code sometimes fails, sometimes passes private test cases).
Nil.
VERY DIFFICULT, QUESTIONS WERE TOO HARD AND GRADING + MARKING IS STRICT
Seems like alot of content being taught from diff areas
Exams can be quite difficult
The problem sets were difficult to code, often taking many hours to days to complete it
Too many coding assignments even if they are required for learning
too fast paced
NIL
nil
The materials are very random and unstructured and disorganized– theres no prescribed textbooks, or lecture notes like in CS2030S. The only way to review contents is slides. And the slides are messy with GIFs, memes, emojis, jokes and not professional at all. One lecture slide file contains around 200 slides cuz there are too many duplicates and pictures. It is very hard to review the lesson contents. Even AI stuck when i send that large file. They should follows how CS2030S lecture nodes works. QnA – we can only ask questions on Coursemology platform and it does not have anonymous mode which makes asking question very intimidating since everyone can see my question
None
NA
na
The graders provided are kinda opaque. Usually graders will tell you WA, TLE, RTE, CTE, with some sample test–cases you can verify yourself. On Coursemology the graders just tell you "not passed" and while the test cases are technically public, they are bundled up so it's like 100 test cases together, which is impossible to verify as a human. We need at least one verifiable test case so we can at least tell whether our output is in the correct format. (e.g. instead of an "empty–maze" test case which is actually 100+ test cases, have an "empty–maze start at (0, 0) end at (3, 3)" test case)
Some additional introductory topics were not covered such as Omega/Theta notation and the Master Theorem which could have been useful in working on later topics.
.
I would like to be able to use various APIs a bit more maybe.
tough tough tough
The lecture quizzes are discourages actual problem solving but encourages using tools like AI due to the nature of grading (no second chances to improve on answer, requiring first try perfection)
online assignments too much java
nothing
Written notes e.g. in LaTeX would be nice, the slides have nice animations but might be hard to look through. Alternatively, providing a reference text as an alternative way to learn could be good.
There are not enough practice materials for exams.
The format of the midterm exams could be improved, if the point of these exams are to test understandings without an actual practical exam, then these exams could have been done in examplify to be more conducive to students. Writing code on paper with all of the constant adjustments required is very time consuming, and being able to type out answers in plain english could save examiners a lot of time as well since handwriting would not get in the way of grading.
NIL
This is definitely a biased take since I was a victim of the mark reduction, but perhaps the capping of the 12m question just due to using the array instead of a hashmap was a tad bit too harsh. Maybe this may stem from my ignorance, but I would imagine that given the rest of the implementation of the solution is correct, making use of the wrong data structure in an interview can be easily

Comments
circumvented by a question on "are you SURE this is the most efficient solution?" or like, some form of reminder that there may be better solutions. In a group work setting, not everyone has 100% of the solution ready in their minds, but one would not discredit the person who has come up with 99% of the solution by saying that "oh, but here you should have used a hashmap instead of an array". Both are valid inputs to contribute to the solution of the problem, and maybe both sides should be awarded with the appropriate amount of recognition that they each deserve (in this case, maybe the other 99% of the solution can be awarded with like a smaller penalty of marks? instead of being equivalent of throwing the whole question). That being said, I do recognise that there are reasons that the answer key had been structured this way (we should still strive to be 100% correct after all), and I humbly accept the mistake that I have made in the exam. I just felt that for future batches such penalties can be less harsh such that there is still some form of differentiation between people who could solve the other 99% of the solution (which does require some creativity) and maybe some who only made it 30% of the way through.
A bit tricky for beginners due to shifts in algorithmic thinking.
my TA was not great for tutorials
Programming exercises could be in python instead of java
Using java yet not caring about abstraction and performance of the language itself How absolutely useless the coursemology site is. The grader is non deterministic, submitting the exact same code can get me wrong answer or correct answer. This grader is complete garbage. I've had the displeasure of resubmitting the exact same code for over 4 hours to finally get it to pass once.
nil
The examination format has been a bit jarring and difficult to adjust to, I wish there was more time put into how to properly answer exam questions with the appropriate linguistics and pseudocode as it felt easy to lose most or all marks for a question due to arbitrary reasons.
problem sets seem abit disconnected with the course, id rather do leetcode
coursemology makes us write in java when in exams we just write pseudocode
It is too fast paced
Course was quite inefficient administratively
there's not really a textbook or central place to access information on the content
I would like if there was a better consolidation of notes because having to go back and refer to lecture slides is painful.
This course focuses too much on theoretical implementations rather than actual practical use. There is a difference between what is best for theory and what should be done in a practical environment.
Recitations are very dense, quite hard to follow.
nil
the speed was too fast
More practice materials for midterms (and hopefully finals) could be provided Why are the component weightages hidden. I prefer knowing even though I put my best in the sessions anyway
nil
HARD
Recitations, tutorials, and lecture difficulty of content feels very disjointed
High intensity Fast paced course Some topics were not explained thoroughly
Very fast paced and lots of assignments
nil
Really jam packed with content especially after mid terms, quite overwhelming.
NIL
Not very used to Coursemology as a platform
I suck at logical thinking and problem solving
Nil
abit fast, content feels rushed
Non–deterministic graders on coursemology, terrible course admin
Tough, hard to grasp...

Comments
Workload
I just find that the curriculum has too many lectures and recitations for us to attend, in addition to a terrible amount of work to do every week when uploading onto coursemology.
It's really difficult and the semester is insufficient time for me personally to master or even fully engage with the content taught in the module
nil
The Lecture notes all over the place, and is preferred to all be compiled together on a website.
Past year papers for midterms were not enough.
Nil
Release weightages pls
–
Some concepts were quite tricky to pick up, general workload was fairly high as well
nil
coursemology is not the best site (tbh i cant think of alternatives but it gave me headaches sometimes)
I need more optional practices, I need more things to do for this course, I want to have to write more algorithm code for the course instead of in my free time for my own sake, add more coding practices, get us to write out each algorithm and compare them in–real time for the same randomised unsorted array. Also, the midterm makeup this semester I took had so much errata in it, it damaged the formality and image of CS2040S, so might have to look into that.
I dislike the hidden grading/weightage of the course (which was later revealed?), while it may serve to let students focus on learning, it may also cause students to be hyper fixated on every aspect of the course, because we don't know what counts, the safest bet is to assume everything does.
How the exams were marked and the explanations/ points we were expected to have in our answers are not exactly clear, especially when it comes to the standard of explanations expected, sometimes it can be a little hand–wavy while other times they are very particular about certain things in responses. Thus, there is uncertainty and I am a bit lost in how to improve my scores for this course. While grades should not be the whole point of this module and learning should be the priority, it is hard to ignore the grades especially so when this is a core module that one cannot S/U.
what is expected of students in examinations is not very clear, what is "sufficient explanation" and "vague" is not clear for pseudocode
too hard
Not being able to use the tests for problem sets after I've already finished them. I want to be able to try out bonus parts even after we can't earn xp from them. They sound fun to implement but I'm usually unable to do them in time for deadline...
Traumatic mod where I feel I am preparing to technical job interview questions as a Y1. I feel I will walk away from this mod feeling demoralised about my major and it definitely destroyed some confidence in me that I could learn anything. There is a difference between code implementation and writing high level code ideas for solution, so it felt challenging to keep up with the theory and the practical via coursemology. I really dislike the assessment format in Midterms (which will motivate how finals will look like) since my entire grade is dependent on if I happen to know how to solve a small number question correctly. I could know a lot but if that question just happen to be something I don't know, my grade will suffer. Its easy for course coordinators to say grades don't matter and its the learning outcomes, but when the difficulty is tuned so high and many opportunities in SoC gatekept by GPA, it felt like the stress and anxiety produced from assessment takes away any meaningful learning experience the mod has to offer.
NA
Too hard
Could give more practice questions. Sometimes the learning curve is very high
the harshness of the answers, its all or nothing, and messing up feels like its over for the rest of the course. the unreleased weightage, maybe cosmology can be kept but i wanted midterms and finals weightage to gauge my position if i can redeem myself in scenarios where i didnt do as well at the start
Most of the Coursemology problem sets felt very unintuitive and it felt like it did not really help me learn or practice the concepts taught very much. The concepts themselves are pretty hard to follow personally, and that sullied my enjoyment of the course a lot, since the concepts were went through very quickly. I do feel lost all the time during the course.
The questions could have been categorised in a more structured way for each topic, such as by different question types, common cases, and possible approaches. This would make it easier for us to refer to them when tackling similar questions in the future.
My midterm not high

Comments
the lack of a clear structure
If we are assessed on pseudocode. Then why isn't there a dedicated pseudocode conventions lecture to teach us how you expect us to write in pseudocode. I have been marked "unclear" in my midterms because of the way I wrote my pseudocode. Which is not my fault because I don't know what you want. I have only seen a few examples that the lectures have shown in class hastily.
introduction to java
No
its hard
there is a lot of content each lecture was 100+ slides and 2 lectures a week.
the workload is crazy
I feel like a bit lack of practice that gives feedback.
The course is quite fast pace and every week we are learning quite a lot of content on top of lab and recitation and problem sets (slides are over 200 pages long)
Hoped to get more extra (optional) questions in LC or Kattis that directly apply the concepts taught to make it more concrete
nil
Difficult
The fact that there were no "FULL" lecture notes first (not the slides, but like a textbook like pdf that we can see and look through), so that we can do a bit more pre–reading if we wanted to, and would give us more independence in taking charge of our learning speed, as these concepts take different amounts of time to learn, based on person to person.
–
i hate now the assessment (midterms finals) are just writing english? im not very good at explaining my thought process in words but i can translate it into code very well. i wld prefer if a larger component of the assessment are competitive programming style, 2h with ide and problemsets.
i dislike the fact that grading percentages are not anounced, while i get the point i truly dont know how to feel and gauge about my understanding in this course
Nil
too hard
The problem sets were annoying to complete because the public and private test cases were too vague, and I spent a lot of time debugging useless edge cases rather than solving the problems.
The shift in pace and content in the second half is rather jarring right after midterms, summarial slides linking every lecture may be a hassle but the benefits for next cohorts will be huge
NIL
sa
Nothing
i want the release of assessment percentage
I feel the slides can be more up to point and provided with summary notes. I feel the weightage is mostly on exam and since we don't know the distribution it does cause more stress. I think could have assignments to actually understand, discuss with peers and learn through projects.
I think the presence of lab and live practical is also important. Like most coding interview use on site coding.
Very difficult course for people new to cs
NIL
The lecture slides are very non–study / non–revision friendly, as its packed with words and conversation style content, not a clear, formatted structure that we can clearly follow to revise.
everything
–
The fact that even if I have a valid and appropriate solution to a problem, I may end up with little (or even zero) marks simply because it's not time efficient. Understand the important of time efficiency though!! Will do better next time.
the marking for midterms was very harsh, which I can understand. but i would like the rubrics to be clear on the reason behinf the penalties. for example, instead of saying –x marks for using array, can add on a explanation why that many marks was deducted. it also provides feedback on where we had gone wrong
no practical

Comments
too difficult
nil
Challenging
Its too difficult and the concepts are hard to grasp
Really hard
Organisation of course content could be better. It felt like the syllabus was just a mix about whatever you thought of at the moment, there wasn't a clear progression in the course content. The lecture slides also felt messy at times. I would suggest have 2 different versions of the slides, 1 for lecture use, and the other a more condensed version for individual revision. Dissemination of course information could also be better. Announcements should be made instead of just embedding the information in the lecture slides (e.g. regarding potential clash of CS2040S and CS2100 midterms). Maybe also replicate important announcements/reminders across Canvas, as for many students, this course is likely the only one that uses Coursemology, and all other modules are centralised on Canvas. The structure of the exams could also be better. As it stands, for the open ended problems, the grading feels extremely bimodal. The way how unoptimal solutions are heavily penalised and how other solutions which might originally have the right intent but mess up the later parts get regarded as being entirely incorrect feels adversarial to the learning experience. I understand the intent is to develop problem solving ability, and these solutions are maybe considered not solving the problem, but it feels overly hostile. I think a design where the open ended question are more structured, which appropriate "stop points", to assess understanding of crucial parts of the question would be better over questions where you have to design a solution entirely from scratch. This would also more closely align with your philosophy of making the tests more true to what we'd experience in a technical interview. In a sense, this structure would kind of be like the interviewer giving real time feedback on your solution, rather than just watching you flounder and give up.
Difficulty is a skill–cap, but then again everyone in NUS is a smartypants.
the problem sets' explanation of problem were really convoluted
The pace of the course.
Just throwing algorithms and data structure at us with no real guidance on how to write our answers in a desirable format
nothing
You get it you get it you don’t get it you don’t get it
I understand the coordinator's intentions but I really despised the opaque grading scheme; students don't have 67 hours a week on one course to perfect every single detail of their coursework not knowing how much that additional layer of perfection would contribute to the overall grade if at all.
a bit too hard, should be easier lol.
why the tutorial questions are so hardddd
Errors in teaching or assessment materials
Problem Sets
Grading is ambiguous
Tutorials and Recitations
The private test cases on coursemology is very annoying as it unecessarily forces me to ponder too much on trivial test cases. But other than that is good.
The marking scheme of the course can come across as very unfair, for example if u get how to do the question but accidentally iterate through the wrong thing with no time to check u can drop from upper quartile to below median just because of 1 question. So it seems like the examiners are not really testing your understanding but how careful u are which can be quite discouraging.
The exam papers are different from past years so it is difficult to gauge, weightage is not known to students as well. papers have last few questions with alot of marks making the paper difficult to score in general
too difficult
The grading for exams can be a little ambiguous and unpredicatable, because of the nature of the answers expected
The assignments and exams are very difficult. I feel not prepared to write algorithms using pseudocode because the problem sets usually use coding.
nothing

Comments
Coursemology public test cases can be more enlightening (which specific test failed and expected output). Sometimes 2 different tests on all possible inputs are clumped into one test case and it’s difficult to figure out exactly which edge case was incorrect
Coursemology
Java
coding portion feels rather complex and difficult given that the course is more focused on theory
Subjective grading and vague constraints. Unsure how to score...
N.A.
It is too difficult, a lot of data structures are taught. During lecture, we are taught how to build the data structures but in the exam, we are required to implement the data structure, which are totally two different concepts.
problem sets and recitation are too hard
nil
The algorithms are too difficult to understand, as they will be very different in the examinations
Problem sets were quite hard, maybe because it is java. The lecturer even said he dislikes java but course coordinators forced java.
Its so hard... I feel like the mcqs being such a huge weightage that each wrong mcq just eats up my mark. How to catch up liddat :(
A lot about graphs...
na
nothing
Very limited coding practice. I find myself unable to implement code during OA because CS2040S didnt teach me how to actually code. Problem sets are quite useless. Language should change to python instead of java. The mod needs a rework.
Midterms mark distribution and grading scheme felt a bit brutal, but can't be helped i guess
Content heavy and quite abstract at times
I think the problems sets are not a good practice for what we need for this course. i think i would prefer questions like finals and midterms with psuedocode for the problem sets.
too difficult, for beginners who just touch programming in y1s1 it is really tough. grading sceheme is brutal as well
The problem sets are too difficult at times
Quite intensive in terms of content and pace
Maybe more clarity would be given on coursemology exp
its hard
No
pseudocodes can be hard to write without clear guidelines
Exams...feels like it ruins the fun of learning the algorithms, but I guess it is necessary
The proofs and mathematical side of things is slightly confusing
none
Nil
NA
Exams are very hit or miss. Large percentage and time pressure on problem solving questions combined with ambiguity in understanding, writing and requirements for acceptable answers make the exams particularly demoralising to take at times. Writing, explaining verbally and typing are very different especially when handwriting tends to be less structured and readable, would prefer a digital exam if possible.
Oops i put under eldon comments
I vehemently dislike the lack of inclusivity in the style of assessment used. The heavy emphasis on examination–based assessments over hands–on, continual learning for grading marginalises those who struggle with traditional examination formats, such as neurodivergent students. Students with ADHD and/or clinical anxiety disorders particularly often struggle heavily with information recall under time pressure, which is exacerbated by the complexity and breadth of the content covered. They also frequently experience mental fatigue and slowed information processing due to difficulties in focusing on precise details. These factors make traditional examinations poor methods of assessment for a portion of students and unfairly segregate them despite their grasp and knowledge of the content. Moreover, I disagree with the decision of modeling examinations after job interviews and believe that it is not an ideal method of testing students' understanding of the content. Job interviews assume a level of expertise from the applicant and often focus on specific problems to solve; this is not the case in this module as students are in the progress of developing their computer science

Comments
foundations and the topics covered are broad. In my opinion, the current system filters in a select–few elite and punishes students who have only begun to fully understand the content but have yet to master them. The system also places an emphasis on cramming due to its content–heaviness in a limited timespan, which is counterproductive to the commitment towards promoting learning by doing over rote memorisation. This further disadvantages neurodivergent people who may struggle with short–term retention. Therefore, I would like to suggest a larger emphasis on continual, hands–on assessment and less on examinations. I would also suggest for examinations to be split into separate assessments; quizzes focusing on questions involving content–recall and multiple smaller tests and assignments to promote active learning. Additionally, delegating tutorials into presentation sessions can help students to share and self–access their learning progress. I greatly enjoy the lectures and the content taught, and would be grateful if the style of assessment can be improved to be accessible and inclusive to more types of students, such as those with the ability to excel but process information differently.
workload
nil
This is a very "either you get it or dont" module.
Feedback for midterms takes forever so its hard to learn from mistakes.
NA
Little bit difficult and prof speaks too fast
Perhaps there could be more guidance for us in the acutal implementation of code (more use of API) etc
high difficulty, unable to test if our algorithms work when done on pen and paper
fast paced
– the exams can feel quite scary because the grading seems to be too strict – at the beginning of the course, some of the problem sets were frustrating as my code failed the hidden test cases (edge cases) and it was very difficult for me to figure out where I went wrong, as the test cases were hidden. (but the AI help feature that was introduced later in the course was very useful)
Too hard for the average CS student.
The concept were sometimes too abstract and content past by too quickly
difficult to "study" for exam, feels more like an iq test. heard the make up midterms was also way easier than the actual midterms
–
too hard
It can be hard learning so many algorithms, and knowing which to implement can be difficult too
NIL
NA
The weightage of each component is not revealed.
Very content heavy.
I think more problem sets that span the different data structures would be better such that we can have some experience in writing code for the different data structures and algorithms
its hard
some of the pses were very tough esp with some private test cases with edge cases I could never really figure out
too fast too tough
too challenging
It still uses Java.
nil
n.a
It is extremely fast paced, but there's nothing to do about it since there is a lot to learn.
it was too much content on sorting
NA
Some of the Coursemology problem sets can get overly tedious and hard to think of from scratch without looking up for some hints or relying on some lecture solution structure as a guideline for solutions
It is important, but still its just really difficult.

Comments
General difficulty of the course and each of the exams
Coursemology is a bit bugged once in a while but otherwise fine.
That lectures are not available as Zoom online lecturers
Maybe it is quite challenging, and sometimes it is unclear what high level idea means in the exam, I'm not sure how much detail to go into
Overwhelming amount of knowledge
Most of the concepts seem so abstract, and saying 'just tweak it like this' or 'modify/change' it is very hard to follow. For someone new to most algorithms taught in this module, this module is extremely painful and the learning curve is extremely steep. It feels as if the content is being dumped on me twice a week, and I am expected to know how to change the different algorithms immediately. It feels extremely discouraging to always be stuck. Since little guide is given on how to actually solve questions, I think midterms are an extremely helpful juncture to see whether the way i write pseudocode is understandable enough, but it is currently week 11, less than 4 weeks to finals and yet I do not get to see my paper. If this is the case, I don't understand how I am expected to improve in such a short time. Additionally, I do not think the problem sets help at all, since there are no practical exams. I think replacing problem sets with something besides the tutorials to enhance our thinking would have been much better, since coding takes a lot of time. I do admit, Eldon is good at teaching, but the way the content is taught makes this module extremely unlikable.
A bit fast paced at some points
the fact that the grade weightages of components are hidden from students. Understand the rationale of incentivising students to not game the system by doing so, but in my opinion there is no need to do that. Anyone who games the system is gaming themselves anyways
The difficulty also meant a lot of time was spent on this course, almost too much time was spent preparing
too many workload(?)
I think problem sets are difficult to approach without the right instruction on implementation (especially since high level ideas are the focus of the course), maybe the problem sets can have more scaffolding in terms of which structure to use for implementation.
Hella difficult and content heavy
nil
i dont like the problem solving i think its so irrelevant! i waste time trying to bridge content with syntax! why? isn't 2040 js logic? then why not stick to more practice on that? there are plenty of other courses and the internet for us to finetune how to translate logic to code!
The exam format: it punishes people who go for optimal algorithms instead of naive ones (former usually get less marks than the latter) In a practical setting, one can try for the optimal algorithm and go back to the slow one if it doesn't work. Here, on paper, it's a gamble whether the true algorithm is correct or not, thus a gamble of a lot of marks as well