COURSE REPORT

Course	CS2040S - DATA STRUCTURES AND ALGORITHMS
Academic Year/Sem	2024/2025 - Sem 2
Department	COMPUTER SCIENCE
Faculty	SCHOOL OF COMPUTING

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

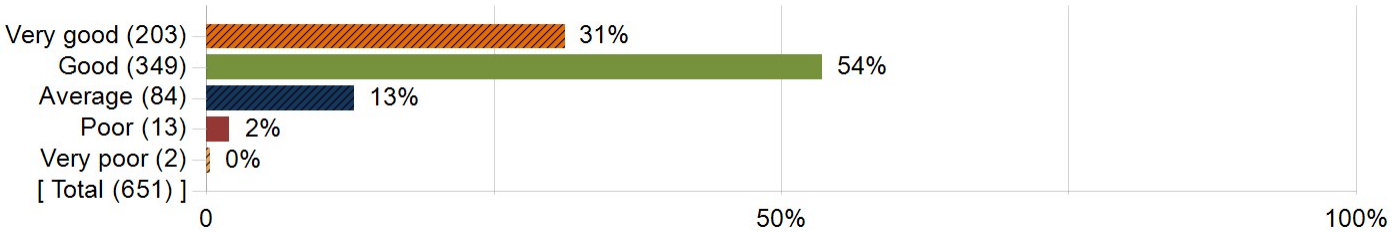
Raters	Student
Responded	653
Invited	808
Response Ratio	81%

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

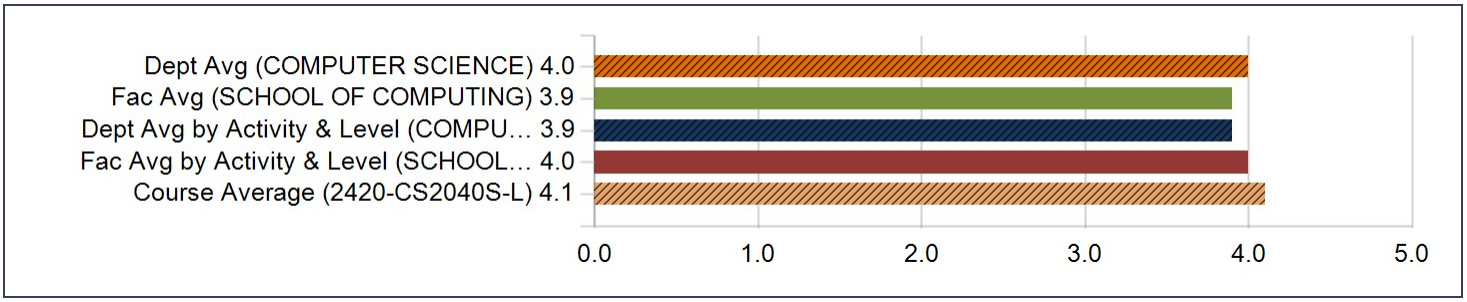
What is your overall opinion of the course?



Statistics	Value
Response Count	651
Mean	4.1
Median	4.0
Mode	4
Standard Deviation	0.7
Positive Feedback	85%

Rating Scores

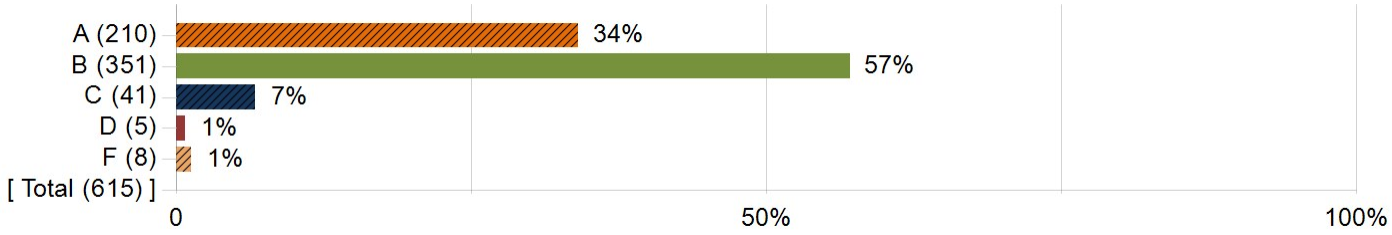
Question	Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))		Course Average (2420-CS2040S-L)	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
What is your overall opinion of the course?	4.0	0.9	3.9	0.9	3.9	0.8	4.0	0.8	4.1	0.7



2. Expected Grade

Distribution of Responses

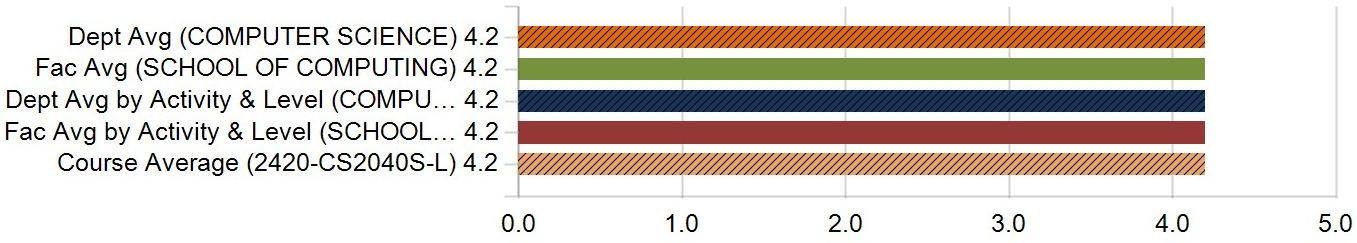
The grade that I am most likely to get in the course is:



Statistics	Value
Response Count	615
Mean	4.2
Standard Deviation	0.7

Rating Scores

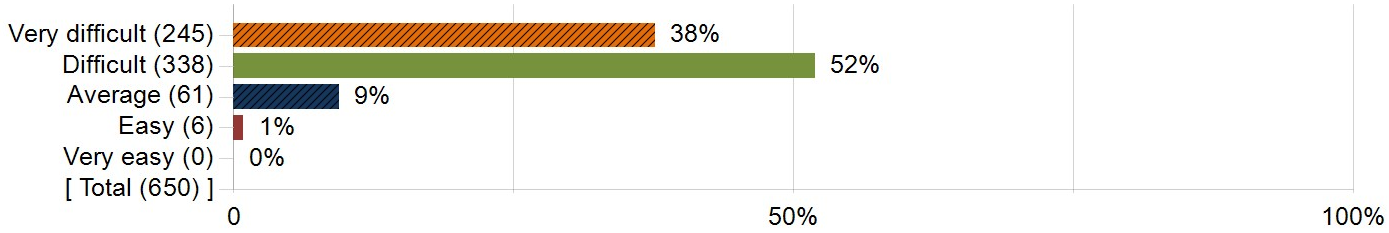
Question	Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))		Course Average (2420-CS2040S-L)	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
The grade that I am most likely to get in the course is:	4.2	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

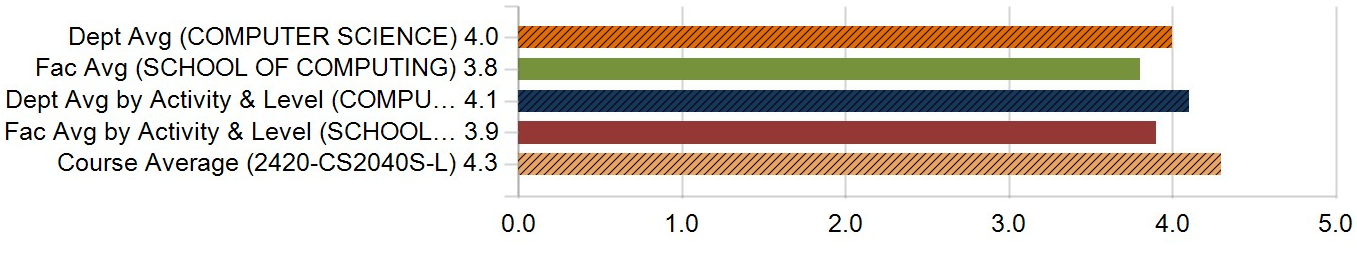
I rate this course as:



Statistics	Value
Response Count	650
Mean	4.3
Standard Deviation	0.7

Rating Scores

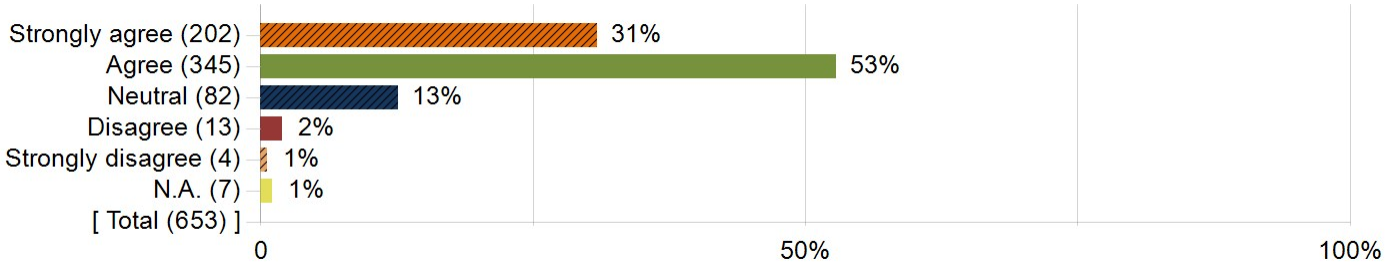
Question	Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 2000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 2000))		Course Average (2420-CS2040S-L)	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
I rate this course as:	4.0	0.8	3.8	0.8	4.1	0.7	3.9	0.8	4.3	0.7



4. Technology integration

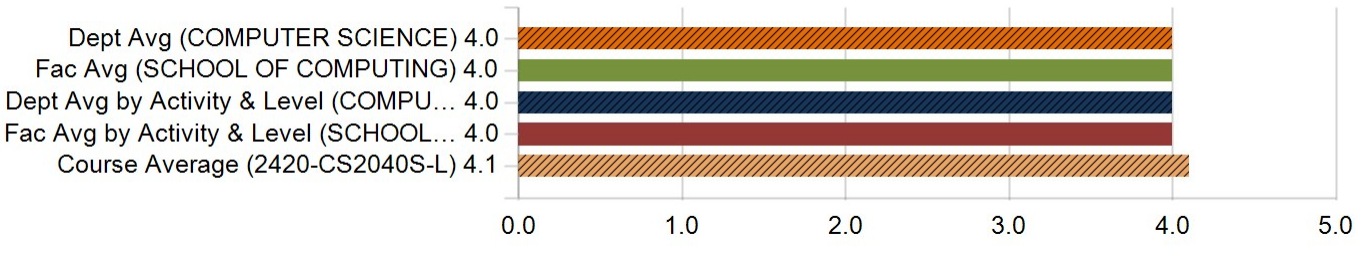
Distribution of Responses

The integration of technology—blended learning, digital tools, AI, and online resources—has enhanced my learning experience.	
Statistics	Value
Response Count	653
Mean	4.1
Standard Deviation	0.7



Rating Scores

Question	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))		Course Average (2420-CS2040S-L)	
	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	4.1	0.7



WHAT I LIKE / DISLIKE ABOUT THE COURSE

What I liked about the course

Comments
Interesting content
Problem sets, tutorials were fun. I loved problem solving, and group discussions were overall quite fruitful. Lecture content was interesting, I genuinely think I really benefited from this course. Both lecturers were effective in their own way, TAs are very responsive to questions, and are very understanding as well. Moreover, I like how the tests are more geared towards conceptual

Comments
understanding, and less nit–picky over specific implementation. Overall very enjoyable module.
The gamification of the problem sets integrate with the concepts for the course make the course more interactive.
EVERYTHING one of the best courses I have taken so far, very professional teaching team and very good.
It features a clear structure, and has so many interesting problems and algorithms to analyse on
Responses to Coursemology forum discussion questions are really fast!!
no comment
Able to explain the basic concepts needed for future courses well
nil
Pretty rigourous introduction to data structures and algorithms
Comprehensive coverage of frequently used data structs
was fun and interesting
Interesting and useful content taught
PSets
Given how important this module is, I came in expecting to have a extremely difficult time but the teaching team broke down the concepts really well and I could understand without much effort
I like how they gave a variety of tutorial and recitation questions to practice our skill on.
–
It teaches u how to think
algorithms
NIL
I like the content
How much we learnt.
I liked learning about algorithms and it really opened my mind to things I'd never thought of before. I am finally able to appreciate the importance of data structures and algorithms in a computer scientist's work.
Useful. Well run.
I think it was interesting mod with learning new things.
generally good teaching team
This iteration had a heavier emphasis on application, as evident the setting of open–ended questions. I really like the way concepts are tested where it is not mere regurgitation–it requires deep understanding of algorithms which the lectures and especially recitation enforces.
dsa
changed my way of thinking when encountering a problem
Learn many algorithms and enhance thinking skills
Eldon
expose to me many relevant algorithms that i will use in the future
–
The tutorials and recitations have interesting question that promote further thinking of applying the data structures to relevant problems.
Learning effective coding and data structure usage
i like algorithms, eldon teaches them excellently.
The opportunities to use various different data structures and algorithms allowed me to learn new methods of thinking.
nil
There are interesting content covered in lessons, alternative lectures (which are better) are available
I enjoyed learning the concepts/algorithm I learned. Moreover, cs2040s teaches student how to analyze and tackle algorithms which I find it useful for us in the future as we get to see more algorithms.
expands and trains the mind to see problems in a different way
Problem Sets
It was really interesting topic and content

Comments
very interesting and useful concepts
nil
lectures and recitations
Problem sets were mostly fun
i like coding in pseudocode
The content is interesting
The instructor made an effort to ensure all students understood the material
no practical
the lectures by eldon are interesting and the algorithms we learn are interesting for problem solving
the pace of the course is really well planned out
Interesting and new concepts with new ways to solve problems
Exposure to a range of algorithms, opportunities to try out the content hands–on
interesting
Learnt many new algorithms
It prepares me me for real world interviews and teaches me necessary data structures and algorithms as it sets out to do.
It's a tough course but because of that it's fun and challenging, understanding the material is rewarding
Problems are fun to solve and learning barely feels like studying.
Tutorial/recitation problems required creativity to solve.
The topic itself
It is a core mod that will be important for interviews and for my future
Made me appreciate algorithms sooo much more. I love learning new algorithms and applying/modifying them to solve different problems.
it was very eye opening to learn about the various DS and algos. i liked the knowledge
i find the content pretty interesting
It teaches me about data structures, which is very useful for leetcode
interesting problem solving questions
It is fun to get mind blowing experience every week
i liked the whole coursemology will the levels and everything. i think the gamification of progress motivated me to complete my assignments and quizzes on time. At the same time, if I missed out on something, i felt like i still had the chance to come back and that all was not lost.
NA
I liked it's applicability to the real world and also its deep dive into the content
Data Structures
Great intro to algorithms and fun questions in tutorials
Difficult and trains us to think algorithmically to solve questions with data structures
Very good lecturers, tutors, teaching staff etc. Support given was excellent and taught very well.
Engaging problems, fun to think outside the box, dedicated teaching team
Fun material, engaging lectures
The problems presented in this course are interesting and really makes you think. I really enjoyed doing the problem sets and optional problems.
Interesting and useful, test more about thinking rather than coding
That i don't have to write the actual code (like practical exams) and just the pseudocode is sufficient
interesting concepts
ensures learning
The topic is interesting, and in general I do like algorithms.
Very interesting learnign about the different basic algorithms and data structures
Interesting to learn how to optimise algorithms, was very eye–opening to model daily problems into data structures and algorithms

Comments
Good structure of course and focus on application of concepts learnt in the course
interesting algorithms and applications
Introduces different problems of varying difficulty and how we can use concepts learnt from lecture, and transform them to approach the problems –> intellectually stimulating
I like to learn about the various data structures which helps to solve various problems
Lectures are recorded, assignments are given online
NA
Concepts taught in this course is very important and well applicable in the Tech industry. Weekly problem sets gave me opportunities to understand the algorithms better. I love how the problem sets are fun and engaging.
Interesting algorithms and approaches to thinking. More skill–based approach to learning.
profs are passionate and make me want to learn more
coursemology exp system challenging and thought–provoking recitation questions optional practices
A very comprehensive beginners course for data structures and algorithms The tutorials/recitation are very engaging and sufficiently difficult enough to improving thinking!!! The lecturers are very passionate when teaching and engages the students in an effective way for them to learn better
infusion of logical thinking
The lecture reviews were helpful to help me reinforce my learning, Problem sets were interesting and fairly challenging
interesting when i see the solution
i learn about dsa which is interesting to learn
it is interesting t o learn various algo andhow they work
algos are fun
I liked learning about interesting graph algorithms such as Prim and Kruskal and Dijkstra. I liked i can see the application of algorithm and using discrete mathematics to prove the correctness of the algorithm and invariants. Eldon is also very passionate and energetic lecturer that made module much more enjoyable.
Too difficult
The teachers
good flow
The teachers are very helpful and engaging
The platform used for assignments were really nice and engaging(coursemology)
Mostly fun working with algorithms, problem sets were fun and engaging to complete
Very interesting way of looking at algorithms.
There are many resources available for students to seek consultation, and many lessons to enhance lecture concepts.
nil
Interesting course especially I like Eldon so much.
Contents of the course itself is very interesting
i enjoyed using Coursemology
weekly programming assignment to enhance my programming skills
Very fun, it's like solving a puzzle!
Interesting and new concepts
This module is probably the most important one in this semester for CS students. It does a very good job in showing how useful algorithms are and their use in our daily lives. It has also made me more motivated to learn more about algorithms.
exposed to dsa
amazing
It is challenging but fun
content is very interesting and relevant
The algorithms

Comments
Opportunities to practice Java design techniques (from CS2030S) in algorithm building to help write more maintainable and less error prone code
It is very useful for job interviews.
Thought–provoking, useful for technical interviews, taught topics pretty well
Lecture reviews
The problem sets are of appropriate difficulty, and quite fun. Tutorials are fun also.
–
nil
High level thinking
good
Interesting ideas
It is one of the most important course in cs
new concepts
I honestly felt that prof eldon carried the module, no disrespect to the other professors though. It is possible to feel his passion for algorithms even through the screen when watching lectures. Lecturers, recitation tutors and TAs all try their best to nurture the interest for the subject despite the high difficulty of the module. The coursemology problem sets are also well designed to encourage students to use and apply the concepts that they have learnt, and the discussion forum is very helpful for students to ask questions.
The online nature
it is interesting
Interesting ways to solve things.
I liked the various algorithms
is learning about new concepts and algorithms
Interesting problems and topics and helped develop my analytical skills
The problem sets were incredibly fun, challenging and engaging. I really enjoyed solving the questions and I think its a great way for students to learn and apply concepts they have learned.
Good intro the DSA
Prompts us to think further and not spoonfeeding answers. Problem sets are helpful
focuses on the theoretical and crafting solutions process of algorithms, which is a key step in solving real–world problems
Teaching you how to solve problems
Focus on High level concept, introduce us a lot of different data structure that can help us to solve question in future effectively
pacing was very good. I especially like that they gave us a lot of down time during the holidays – Hari Raya, CNY, etc.
I liked the problem sets that allowed me to try implementing the algorithms I have learnt in the lectures, and learning about the solutions that are used to solve difficult problems is very mind opening
The interesting algorithms and data structures taught.
It was quite interesting to learn all the theoretical algorithms.
very interesting but difficult
very relevant for aspiring software developers, important to learn and pacing was good
that i can use pseudocode
The amount of knowledge I gained from the module and the structure of the module, but this is only because I was able to do rather well for Mid–Terms, if I did not do as well, my opinions on the structure might change. The weekly problem sets were also rather engaging and fun to do but the private test cases on coursemology was a little annoying to deal with.
alot of content and stimulates thinking
algorithms are interesting
Interesting algorithm
learned about some cool algorithms
It's an important foundational course to computer science in general.
nil

Comments
We mostly work around pseudo code which reduces the time spent on fixing bugs.
Generally progressive nature of the course, lectures are often the hardest part, then recitations and tutorials but problem sets are good to learn from and exams test manageable content. Lecturers oftentimes put the effort to explaining the relevance and importance of the course for our future job/ internship hunting. Course uses java as a tool to teach core concepts and algorithms, rather than overly focusing on the language itself, ensuring most of the times students can focus on the understanding of the algorithm rather than wasting time focusing on the syntax of implementation.
As per mentioned, the lack of focus on grades is relaxing
The various lesson modes (lectures, tutorials and recitations) are all highly engaging and reveals crucial insights in terms of content and thinking process to us. The learning assignments (lecture reviews and problem sets) are also mostly engaging and enjoyable, helping us to review and practice what we've learnt, as well as write code to solidify the algorithms we learn. Finally, I appreciate the move towards open–ended questions for our written exams, rather than convoluted MCQ questions that end up filtering students by meticulousness, punishing careless mistakes rather than rewarding high–level thinking.
none
math
It was very challenging!
Thought provoking and forces students to think of problem in a different light
interesting algorithms introduced
Exposure
Helped me get a good umderstanding of data structures and algorithms
Concepts taught in class was interesting and relevant to critical thinking and problem solving. Workload is just nice, not too much and not too little.
Eldon
The gamified system on coursemology
I like the problem solving aspect of the course
Taught me a lot of and about algorithms
Enjoyed the tutorials, very interesting questions. The programming assignments were also a great way to keep students caught up with the teaching.
–
Cool
I liked learning the different algorithms and data structures
Problem sets and recitations are really fun, content is definitely useful and interesting as well.
Coursemology
Learning creative ways to solve problems
interesting concepts
The lectures are amazing. I thought i would not enjoy this course, but Eldon made it so much fun.
capable of learning algorithms to solve real life problems
tutorials and problem sets were fun to do
Its a very useful course for my own development.
The algorithm concepts are very interesting and taught in a rather structural manner
It covers quite a lot of topics that I have never seen.
the problem set
The teachers and teaching assistants were generally very good and there was a lot of support provided.
I like that it enhances my thinking skills and it is a very important course for my future
Moving away from pure MCQ exams
Useful
I get to learn and better understand algorithms
The course is very interactive and gives us many opportunities to use what we have learnt to solve real world non–trivial problems. The course also provides many opportunities for us to clarify our doubts and get support.
Provided intellectual rigor

Comments
Algorithms
Algorithmic analysis, new ideas, new techniques
Problem sets and exercises are fun to do.
Really fun way to tackle thing, the PSETS are so fun
coursemology is fun
I liked the fact that it is rigorous and covers a broad range of content that is extremely relevant in developing our knowledge of algorithms
Coursemology as always is great, provides the much–needed practical aspect to DS&A
NIL
The concepts covered
nilnil
Problem sets and tutorial questions throughout the semester which allows students to think critically and learn more about DSA in an interesting path.
Interesting concepts and recitations and tutorials show how they can be related back to real world problems well
How we can use different algorithms to solve many problems
seems quite helpful
nothing
Content covered is useful and interesting
very interesting
Tutorials and recitations have some interesting questions
It really teaches you on how to think about data structures and algorithms, achieving exactly what the module is supposed to do.
nil
The coding exercises were quite helpful to put the concepts in practise
More freedom and allows us to convey ideas through pseudocodes instead of correctness.
useful content
quite fun course
Eldon
algorithms that are showed to us and those that we get to interact with and actually use are very fun. the psets are fun to do and the algorithms and data structures that we learn are fun.
I enjoy learning algorithms
NA
nil
The problem sets are fun to solve and of the right difficulty—hard enough to allow us to apply our knowledge but not too hard that it is impossible to solve.
That the lectures were fully recorded because i really need to keep going back to see the previous parts if not i cfm fail bruh
informative and intuitive
cool concepts
Good blend of online and in person, intellectually stimulating
NIL
nil
It went through important content
Eldon :)
Algorithms are very interesting to learn.
Interesting subject, thought–provoking
The content is very essential to CS as a major and I like learning about the content when it's presented in an engaging, relatable manner
Interesting algorithms and problems to think about. Workload is just nice too.
na

Comments
learning different interesting algorithms
Lectures being available online, course was reasonably well structured and content was interesting. I liked the format change of the midterm (and exam) away from MCQs, but I think the open–ended part should be shorter / worth less marks as making a mistake in the initial part of the question is too punishing (and I don't think being extra lenient in marking there is the best solution).
algorithms are fun
Encourages me to think outside the box for modifications to algorithms to solve different problems more efficiently
1. The problem sets. I think they really helped in understanding the practical application of the concepts taught. 2. Eldon's lectures.
I liked the problem sets, they felt very manageable and gave me useful experience in coding out the actual algorithms. I also like that we use pseudo–code for actual pen and paper exams because writing out code by hand isn't fun, and the idea is more important than the syntax for this course anyway. XP system is always nice, feels more friendly and fun. Course workload doesn't feel too high, just about right. Lenient half xp policy for late submissions are nice also.
very intesting take on algorithms, also with the psets it did make algorithms analysis more fruitful and insightful.
helpful
The content is interesting and stimulating
The slides are easy to follow after the lectures when I am trying to make notes for each chapter
The content
it is quite helpful in helping us develop problem–solving mindset and skills
focuses more on the idea and concept rather than the specifics, which better encourages one to think about the algorithms more rather than focusing on the specifics of the code
I like the content.
Giving us a fun problem set that concurrently provides a space to implement knowledge obtained from lectures as well as a glimpse of real world problems.
Content is fulfilling
The algorithms learnt are interesting and really provodes my thinking.
Nil

What I disliked about the course

Comments
coursemology is down multiple times and it affected my submissions. The lecture review opening timings and bonus was very random and not exactly after lectures. The lecturers were difficult to understand.
Workload tends to be a little heavy
The fact that there are only 2 major tests makes it very stressful to prepare for. I would love for there to be a practical portion to act as "insurance".
Nil
NA
sometimes the pace may be a little bit too fast
I have a few issues with Coursemology: 1. Auto–grading is a double–edged sword: I think is discourages us from thinking of edge cases and testing our code ourselves. 2. I think the leadeboard should be abolished or removed. My understanding is that CS Education Research literature has found that CS1 courses have a so–called "bimodal" distribution* (see Robins, "Novice Programmers and Introductory Programming," in *The Cambridge Handbook of Computing Education Research*, edited by Robins & Fincher, pp. 332–333). I know CS2040S is not a CS1 course, but I suspect it has a similar grade distribution. My concern is that students who struggle more, not due to lack of ability or effort, but simply because they have less background, see high performing students on the leaderboard because they have more computing background, end up discouraged or feeling inferior. This kind of ends up discouraging students from focusing on their own learning and improvement. * I know the term "bimodal" in this context is arguably misleading/harmful, but it suffices for the purposes of this comment. I'm not sure why this course never really properly teaches Linked Lists or what the greedy algorithm is. I did not take CS1101S/CS1010X, and am privileged enough to have previously learnt these topics, but my friend who took CS1101S told me he was not taught about linked lists or greedy in that course.

Comments
The slides have a bunch of As someone who actually reads the textbook, there is some difference in implementation/terminology between the textbook and the lecture content (e.g., binary heaps in the textbook does not use a hash table for constant–time contains(), the textbook commonly uses the tilde notation not covered in class). To be fair, I think the number of students who actually read the textbook can probably be counted in one hand Also, I think the course's AI policy violates NUS's overall AI policy (https://ctlt.nus.edu.sg/wp–content/uploads/2024/08/Policy–for–Use–of–AI–in–Teaching–and–Learning.pdf), which states that "If the decision is that students should be forbidden from using AI tools for an assessment (for pedagogical reasons), then crucial aspects of that assessment should be conducted in–person and instructor–supervised, to ensure that students do not access those tools." (p. 4). Nonetheless, I agree that using generative AI tools for the Coursemology problem sets is not good for learning, and recognize various challenges in having supervised practical assessments for this course. I think a potential solution would be to have tutors ask randomly selected students to, in a supervised setting (that is, in person), explain segments of their code, which they are likely to be able to do if they coded themselves, and unlikely to do if the solution was wholesale/largely copied from a generative AI tool's output. I think the pen–and–paper exams focus more on application of the data structures & algorithms taught, rather than their implementation and proving their correctness/complexities, which feels weird because that's what the main part of lectures are about. I don't deny knowing how to apply these data structs & algos is important, but my concern is that then students may start treating these like a black box without properly understanding how they are implemented.
no comment
The tutorials and recitations are difficult
nil
A bit more theoretical than I expected, I thought I will be required to implement all of these but apparently not
Would prefer if there was a practical component for this semester
too difficult and i fear for my life that i am going to fail badly
Coursemology deadlines were quite tight
The weightages of the exams were not disclosed, and even though I know it was done so as to motivate us to focus on learning, I still wish they were released so that I could allocate my time better with regards to studying.
–
a bit too difficult
hidden test cases in ps
NIL
Can be confusing at times, maybe more visualisation can be done
Things are often too difficult with insufficient help.
The workload was a bit too exhausting in all honesty. Additionally, while Coursemology is a great platform but sometimes it would be difficult to debug problem sets when only vague descriptions of test cases were given.
NIL
I think the concept though in lecture and classes are not make us well prepared for the exam
Quite difficult and exhausting
problem sets were sometimes too time consuming
Would have been nice if there were 1–2 practical assessments.
na
problem sets are too tough
The submissions (like lecture reviews and problem sets) felt non stop. always felt like I was rushing to catch up on them. Also questions are always very wordy and sometimes not phrased very clearly
midterms last question is unclear
final few topics are very content heavy and unintuitive
–
To hard.
The slides for lectures are very confusing and unnecessarily long. It could be further shortened or the slides without the animations

Comments
could be included for easier referral.
I think the lessons are very rushed ? It feels like its information overloadding at times. Also I felt that it was too theoretical for me, but that could be perhaps I like practical. However I felt that the course was simply focused too much on the earlier parts and skimmed through the later half of the course. It felt that midterms is only tested on 2 topics and now finals is coming up its about to test 90% of the remaining topics ? which I felt was strange
nil
I feel that there is just too much content in this module to properly master all of them of even just know all of them, i understand that there are alot of data structures and algorithms to cover but there could have been more emphasis on the definition and understanding of them rather than the usage
nil
Coursemology, course can be a bit difficult, content can get abstract and hard to understand
It teaches a bit of SE skills with the problem sets. Really torturing and time–consuming to debug TT
dynamic programming specifically
TOO MUCH THINNGS SO BEHINDWTF I TRY SO HARD RN II FEEL LIKE I HV DEPRESSION
private test cases and certain public test cases that are not very useful in figuring out whats wrong.
Not transparent with the grading system Could release midterm results earlier for revision
I think there needs to be significantly more practices. Moreover, I dont like that the grade is rpedictaed entirely on just midterm and finals. There should be more components
hard:(
Too hard
algorithms
Some problem sets and optionals had public test cases that have expected values of 'All correct' which is quite pointless for a public test case as it doesn't really give me any useful information. Maybe the test case that failed could be shown for public test cases so students can have some form of information to know where to start debugging. On top of that, too long exceptions for some of the optionals and problem sets should inform the student, if not they may think there is a logical error in their code and not be aware that the code works, but is just not as efficient as required
i found the problem sets very difficult
The exams seem to be planned poorly. The format was completely changed with no rationale provided and the marking schemes seem to be zero sum , with no partial credit given and with no proper feedback. Due to the new format, no past year papers were provided which makes no sense to me because the questions could at least be used to familiarise ourselves with analysis. The paper that was provided was not even reflective of the actual difficulty level, which seems to defeat the purpose of even doing it. The actual midterm paper was also completely different from the tutorial questions and recitation questions, which primarily focus on concept testing and application, while a huge chunk of midterm paper was writing pseudocode to implement a data structure.
n.a.
private test cases
some lectures seemed very rushed with a lot of content being covered in the lecture
the slides are usually confusing to understand
the difficulty
The problem sets and recitations seem to require students to implement solutions for poorly defined problems, providing no solutions by the end. The lecture slides/content are rushed and have poor flow, which makes them difficult to follow. A significant portion of the course materials are filled with ambiguities and lack sufficient guidance for students to learn from scratch. Much more rigour could be introduced.
hard
The examinations are difficult
NIL
The lecture material is a bit disorganised, making it hard to search for information when studying
I think we should spend more time learning about how to come up with creative solutions.
Coursemology site kept on crashing whenever I wanted to upload new code or test them.
This course (as with other CS mods I have taken so far) doesn't sufficiently *teach* problem–solving techniques, so you have people that know about DS and algos but not how and when to apply

Comments
NA
maybe the format of the papers? feels like if i cant do one question(last question) the whole paper's score will be damn bad alrd
i dislike how i dont know anything about the percentage or mark distribution of the course
nil
NIL
The coding part is not really related to the course and its private and edge case super annoying
the slides are quite bad. modifications can be made to the slides to facilitate revision there are too many duplicates of the same slide (esp when showing how algorithms work), which is quite annoying to scan through etc. a shorter deck of slides for dissemination would be good, as it facilitates better note taking and revision
NA
Lack of coding assessments that could be more useful
the switch to non mcq exams with the lack of practice papers make it difficult for students to practice. students dont know how verbose they need to be in their exams to explain
Too hard and the content is too packed
Public test case should have public verdict so students dont struggle on minor mistakes and waste their time I think a presenting a more united way of pseudo coding for exams so there will be no misunderstandings and lost of marks
where is seth
Personally questions can be easier to read. Eg invariant questions usually have a bunch of letters like i, j, k which makes it hard to understand and process and often becomes a guess to what they actually mean. More CAs as well but that might be cause I did badly for midterms
Some level of abstraction is too difficulty to understand, workload is very very high
nil
Extremely difficult to apply course learnings to problem sets, greatly dependent on problem–solving aptitude that may not be innate to everyone
Very specific but i think the time limits for the optional problems might be too strict. Even with the optimal time complexity I had to submit multiple times to get lucky and have my program run under the time limit. counting inversions >:(
Psets can take really long to solve
Its just how the lecture works(there wasn't indepth explanation unlike the other two)....I learn more in tutorial and recitation than lecture where i sometimes dont get whats going on
nil
nil
I find the problem sets have insufficient explanation, and can be unclear.
feels like too many but cant really blame the course,
Problem sets were really difficult and coding skills needed were not really covered in CS2030S in depth such as the different data structures in Java, so it was definitely a struggle just trying to get working code even if I had an approach in mind to tackle the question.
Administration was slightly messy in terms of tutorial bidding and allocation
for weekly coding assignments, cannot see what specific test cases failed
Can be very tricky
Not sure why our homework is all about coding when we barely touch on coding the lectures. Yes, we are supposed to self–learn but at the same time its too fast paced for us to actually absorb and implement our codes. Also, maybe give a few hints with regards to the failing of test cases as some people spend a long time(4 hours+) trying to figure it out.
Lecture quiz submission deadline is too strict
NA
NIL
no practical examinations, how can we be assessed on our ability to implement algorithms if we are not tested in a concrete manner
lecture video uploads occasionally not uploaded until very late at night, which becomes somewhat problematic if the lecture review deadline is 1 day after, maybe extend it to 2 days? lecture reviews might be a bit too hard, some questions can be quite vague

Comments
coursemology messed up eg two copies of lecture review 18, uneven levelling exp requirements lack of transparency in coursemology private test cases – even if you dont want to reveal the test caes at least tell us if its a CTE/RTE/wrong output/TLE/etc? midterm took too long to be graded (midterm in week 7 and results released in week 12)
Shld include a practical exam if they want to test practical and not in midterms/finals
java
24h deadline for lecture review was a little bit tight
can never thought about the solution
too much content. week after week i felt like i was drowning with content.
it is so just damn difficult for a student that has not been in this field to ace in
Very high time commitment on solving problem sets Bug finding with cryptic public test cases (especially the later problem sets where all we see is "all correct" as the output requirement) and worse still the private test cases can double the time spent
Coursemology
sometimes the content is hard to digest because the slides are a little confusing
I feel the graph algorithm and dynamic programming could have been allocated more time and sorting part be allocated less time. I feel we had rough idea of sorting in 1101s and graph algorithm and DP were the more complex and newer topic. I also wish there was a section about string sorting.
Difficulty
the difficulty
nil
So difficult.
Coursemology hidden test cases and not showing errors
This year's iteration of the course is really poor. Compared to previous years, slides were messy and full of errors. the formatting of the slides uploaded were also really bad. I appreciate Eldon's effort in trying to make it less theoretical by making exams open–ended, but I feel like the execution was really bad. Sample papers were full of errors and the mark distribution of questions are really weird (like 2 marks just to increment advanceday) but 8 marks for a whole dp question?. In general maybe putting a lecturer fresh out of graduate school into being in charge of a module with almost 900 students is not a really good idea.
Please improve the testing on coursemology problem sets. We are sadly not veteran coders but newbies. When a code fails a public test case, we do want to see what went wrong, but half of the test cases don't have a clear description of test and expected result. It just says "All must be correct" under expected result. How can we infer from this? I feel like many have wasted time trying to pass test cases, not knowing what went wrong. I understand you are training us to test it ourselves. However, this is counter intuitive in a time consuming course like CS. Please adopt what CS2030S team did, by actually showing what are the tests and what are the expected results.
The first problem set was given when nothing was taught, and students who did not have experience with java were at a huge disadvantage. I felt neglected and thrown into the sea without knowing how to swim.
algorithms
Nothing only too hard.
Difficult to grasp some concepts
The lectures are a bit bland. Feels lifeless...
the bonus XP is a bit not so good in encouraging students to learn instead of just getting it done on time
For problem sets, there are no public test cases that test invalid inputs. Instead, it would just say "Your code fails one or more private test cases." I think there should be some public test cases that cover invalid inputs.
lecture slides were not very detailed and no supplementary notes were given
I would've wished that coursemology has more basic exercises for us to implement ideas learnt in lectures, with testcases to check for correctness. Furthermore, problems sets could have been newer, instead of using ones from previous semesters.
nil
NA
Workload per week is too much if combined with another CS mod
workload is extremely high, and challenging
Exams are needlessly difficult, felt like the new exam format is not bad of a change as it is more in lined with the demands of

Comments
technical interviews. However, I disagree with the way the open–ended questions were marked as I felt that if a question takes up almost half of the marks of the paper, examiners should be more flexible when accepting answers, and should not completely give 0 marks for answers that exceed runtime or fail to meet a certain demand. Personally, I feel it makes the examination very luck based because a lot of algorithm questions is either you get it or you don't so if it is marked like that, there can be huge disparity to two different students' score even if they both done their due diligence in preparing for the examination and can discourage CS students to study or put it effort. Just my two cents.
Strict deadline on lecture reviews
high workload
The midterms and finals are hard to prep for because there isn't enough similar practice. I also wish that the exams were either fully open book or that they didn't test specific things from the course. For example, CS1231S allowed the testing of any result taught in the course, but to balance it was fully open book. In CS2040S, you might be tested any specific algorithm that was ever mentioned in class, but you are only allowed to bring in 1 cheatsheet. This does not feel very fair, when some answers are entirely dependent on some of these results (like the priority queue implementation in tutorial 7)
–
nil
Lectures can be abit messy at times, and hard to follow. There is not alot of teaching on how to convert the algorithms into code, which makes it very hard to do the problem sets.
good
The problem sets are too tough and time consuming
NA
difficulty
Questions feels like either you know or don't know, no in between. I think that there is no choice though, one can only improve through more practice and exposure to different algorithms. However, this makes it feel either very rewarding or very underwhelming and disappointing (this is more often the case). Coursemology is also often buggy and slow, perhaps due to students' inefficient algorithms though..
Some things are abstract to me
it is very rushed
The deadlines for the lecture reviews are quite short.
the pace was quite fast
workload
problem set public and private test cases input not given
None
Please provide more digestable materials to study and revise in our own time besides the lecture notes. The lecture notes are super messy and not suited for studying.
NA
Mateials keep being given in Wk13, 1 week before finals. Quite ridiculous.
because it focuses so much on theoretical, its a bit jarring to suddenly leap to problem sets, idk maybe have more practical solutions so the leap isn't so weird, or maybe its just me and i have skill issue
Sudden change of exam syllabus compared to past years
High workload and tight deadline on problem set on coursemology
maybe we should have a PE next time. that would be fun.
The content is very difficult and the problems in the tutorial can be difficult to wrap our heads around. The recitation worksheets are also longer than what can be gone through in the class, so it would be better to have a longer lesson for it. Also, the problem sets are very difficult to debug when there are errors, and took many hours just debugging. It would be nice to have more test cases that we can run on our laptop to identify the problems our algorithm has
The marking time for the midterms. Only 3 TAs/lecturer were marking the midterm scripts (around 800+ students) and because of that it took a horrendous amount of time (4 weeks) to get back the first draft of the scripts. I feel that for subsequent semesters this definitely needs to be improved. Course transparency? Reveal the distribution of the course grading like the % of coursemology, midterms, finals for the final grade. Also release the statistics of how students performed for the midterms since I do not think this course adopts a non–bell curve approach, so there is no reason to gatekeep it.

Comments
The problem sets were quite challenging and time consuming. But definitely a great learning experience.
before the mid semester the slides were not very well structured now it is a bit better
workload..
the problem sets were too reliant on coding ability
The nature of the topic is such that questions are very polarising, you either know the solution or you don't, thus the differences in scores will be very drastic and if you had one bad day, your marks on a paper could drop drastically as you could not figure out the solution for one question. I hope more leniency would be given and the questions or marking scheme be less polarising?
a lot of content and alot of pressure to finish the exercises for exp on time
it is very difficult
Too hard
very theory focused, not sure if it'll help me pass technical interviews and i didn't really feel like i learnt anything new that will be useful for me to pass technical interviews in internships
too many quizzes. too many problem sets. deadlines too tight. when doing problem sets on coursemology, the testcases sometimes don't tell you what exactly is failing, and you have to waste too much time debugging.
this course is the most difficult with less practice given to prepare for exams. Workload is too heavy and sometimes unable to catch up with the pace. Problem sets are extremely difficult and too much time is spent on them. The questions asked in the exams is sometimes not even taught in the lecture which is totally unfair.
The problem sets are difficult
Nearer to the end, problem sets are abit too lengthy, possibly could be taught in a more practical language like python? so students will be able to both understand the theory and apply the concepts practically in a language they are likely to use for technical interviews in the future.
I don't like that the public and private test cases for our problem sets are completely hidden from us, which results in countless hours of guesswork in order to find a random careless mistake in my code.
none
coding
The problem sets were bit too time consuming, and while it tested the concepts taught during class, I feel like it could be utilised to test more concepts taught during class!
Coursemology test cases can use larger numbers so that the constants in running time does not cause failure in smaller time complexities when passing the larger time complexities
Too challenging at times, find it an uphill battle to compete with people with prior experience.
difficulty level is high
Weekly assignments were a bit difficult to keep up when combined with assignments from different mods. Not providing the desired output for each testcase for the problem sets which was time consuming if it did not pass the testcases
Deadline for lecture review could be extended to end of the week rather than end of the next day as it can be quite rushed.
nil
A lot of the times the feedback for code given on coursemology is very vague and basically just states whether it passed or failed. It would be nice to get more detailed feedback so I know what to edit
The difficulty of the course is really high
Difficult
–
Hard
The slides used can be a bit confusing at times.
I feel like this course is quite difficult, sometimes coming up with a solution is not very simple.
Wished there could be practical exams maybe? The depths for which theory can test feel a little shallow
The new exam format is alot more difficult
nothing
challenging
Why can't i find the weightage for each component in grading?

Comments
for people who do not know java beforehand and who do not take cs2030s, the problem sets require a lot of online learning and time to complete outside of the course. as the examinations mostly require pseudocode, the need to learn code in java is a little redundant for examinations.
To be honest, the course grading is not entirely a test of how well we understand data structures and algorithms, but rather sometimes it boils down to whether we have seen a similar question before.
The content is a bit too much relative to the lecture hour and sometimes it could be a bit hard to follow up because a lot of things are only covered in tutorials and recitaions(but in a strucutred way),
There aren't that many interesting problems in the tutorials
lack of math, can introduce more algorithms
It is very difficult and none of the grade distributions were released making it difficult for us to balance out our efforts (ideally we could distribute our effort equally but that is not realistic given the number of mods we have with very different difficulties)
The exams are mostly curveballs and basically an IQ test.
1) Confusing use of pseudocode: While I get that pseudocode is useful in reducing the mental overhead of an algorithm, I feel like the use of pseudocode in CS2040S is confusing, because there's a lot of inconsistent usage (e.g. arrays starting from 1 for sorting, but adjacency list arrays and graph nodes starting from 0, etc.), maybe it would be a good idea to include both pseudocode and Java–based implementations within the slides? 2) Optional Practices: I believe the time limits for them should be increased, as there were numerous problems where my (identical) solution passed on Kattis, but failed on Coursemology. It's can get very frustrating trying to optimize the code to make minor optimizations when it should pass all test cases. 3) Course workload: CS2040S is way too heavy for a 4MC course – per week, it has 3 hours of lecture, 1 hour of recitation, and 2 hours of tutorial (that can each take up to 2.5x the allotted time to properly digest and understand the material), not to mention biweekly lecture quizzes and weekly problem sets (that can take 4–5h/week to do), meaning for some weeks I can spend 20h/week on this module. I feel like the true workload is probably closer to 6MC than 4MC 4) Lack of transparency regarding the grading weightages
Course is hard
Test cases for problem sets can show the expected outputs for inputs instead of just stating "All correct". Public test cases can also be given.
Because of the fast pace and difficult application questions, the course highly favours those with prior experience.
Explanations in lectures sometimes too handwavy
For some of the weekly problems sets such as PS7 Part 2, there may be multiple solutions with the same answer. However, Coursemology requires a very specific solution in order for its test cases to pass, but this information is not available on the question sheet. Hence, most of the time spent on the problem set is relegated to trial–and–error testing to guess the rules of the specific solution, which distracts from the main learning outcomes. I hope that in future iterations of the course, the problem sets will avoid this unnecessary complexity.
everything is so tough to understand. There's like not enough time to fully understand each topic.
the deadlines on coursemology kept changing
nil
More practice on difficult concepts.
Contents are sometimes a little to simplistic and explanations are a little roundabout sometimes. Slides are a little unclear at times. Questions in problem sets / tutorials / recitations are sometimes a little unreadable / hard to understand.
na
the release of problem sets seems to be abrupt most of the time, especially the last few problem sets where the date of release is always changed at the very last minute
– Too much pseudocode. Implementation details are often left out, even in tutorial, when tutor is asked about implementation details, he says don't need to bother. But knowing how to implement is important too I feel and should the focus as well. – Public test cases not shown. Private test cases also not shown. This is not good for learning, as we spend hours trying to figure out why, which really is mentally draining and a time sink. – Only 1 hour allocated for recitations, but so much to cover. My tutor hence ends up rushing through the slides because there is literally no time to cover every question. Especially recitation on (A, B) trees. Please allocate more time for recitation so that the content can be taught properly instead of being a rush
None. This is a tried and tested course and should continue being the same. Eldon needs to continue lecturing this module though
Lecture material is EXTREMELY disorganized (again because of hand–me–down nature) and impedes learning. If this continues I learn more from Wikipedia than the lectures.

Comments
Course cannot decide if it wants to be a practical module, or a theory module. It should be the former in my opinion, and heavily augment Coursemo/utilise other platforms to achieve this goal.
NIL
NIL
nil
Workload is heavy although its fun
coursemology feedback for problem sets were pretty terrible, did not help with trying to bug fix
wayyyyyyy too much content, so much, hard to be fully prepared before finals
nothing
The exams' marking scheme is quite questionable. Marks seem to be all or nothing for majority of the questions. No partial credit was awarded for partial correctness even for 7 mark questions. Moreover, there was no response to appeals submitted regarding the marking of midterm paper.
Some questions in tutorials, recitations and problem sets are unclear and not phrased well. Sometimes no response from teaching team or TAs on coursemology when clarifications asked. Questions posed feel restrictive, as if there was already a pre–planned solution, instead of encouraging students to think creatively and propose new solutions for discussion.
too much content. 2030 has alr ended last mon. but here we are still having lectures. seriously please end this. and too much lecture time. 2h a week is enoigh man
Not a big fan of some implementations in the problem sets.
I wish it was more clear about the breakdown of the syllabus, though admittedly it doesn't matter as much.
nil
NA
Too much content
sometimes a little hard
weird way of implement dijkstra algorithm
The problem sets were way too hard, like we are expected to learn how to use hash sets and maps and everything in java on our own and it's quite a struggle
can we have more time to complete the coding assignments (the psets)
NIL
NA
nil
No transparency in grading weightage
The content is not very structured, making it hard to follow at times. The tutorial and recitation sheets also provide little to no guidance on how to approach the problems; they are set in a way which expects students to already be familiar with the thinking process, which is unrealistic since we are learning it for the first time.
too hard, recitation and tutorial material way to convoluted, recitation legit too hard at all like it completely did not help me for my understanding, almost always involves some other higher level concept from another field
NIl
problem sets near the end were abit dissatisfying to do as the code structure was confusing and restrictive in a way that did not seem to be intentional or helpful to teach us the algos
too many concepts
Workload was extremely high and demanding
NIL
nil
Algorithms are very hard to learn. At most of the time, I cannot think of the algorithm to solve the problem.
NIL
na
there is a gap between the content taught and the exam expectation. even we understand how an algo work we may not able to apply it effectively or come up with variances.
Coursemology is a little glitchy at times. Public test cases really need to be public (weird that the expected results / my code's output, or even the test case for that matter, sometimes isn't shown, only the file tested.) The tutorial and recitation on SSSP were

Comments
pretty much the same lesson twice. I think there should be greater transparency in the grading (% of each component, showing medians, Q1, Q3 etc).
sometimes it gets very hard
I personally find it quite challenging, but its more of a personal limitation rather than a course flaw
more sample practice papers would be nice
felt like a lot of stuff were more rushed (like the information is just thrown to you without much explanation) which might make it a bit more difficult for some students to catch on
nil
Slides and presentation of content felt subpar and was hard to follow at times. Perhaps a concise summary / refined notes could be provided for students for doing revision (similar to CS2030S). Requirements for pseudocode are also a bit ambiguous for assessments – more practice papers should have been provided for students to better understand requirements.
The difficulty level of the recitations
nil
nil
Heavy weightage on the assessment components.
The use of coursemology is not as efficient as using canvas
coursemology makes me mad.
Heavy workload, and difficult for me:(
Nil