



# 2022 Fall Individual Report FAS-COMPSCI 121-Introduction to Theoretical Computer Science 001 Boaz Barak

Project Title: **2022 Fall Harvard FAS Course Evaluation**

Course Audience: **101**  
Responses Received: **90**  
Response Ratio: **89%**

---

## Report Comments

Note:

The order that the questions appear on this report is not the same as the way the questions were displayed to students. The order has been changed to make the report more readable.

---

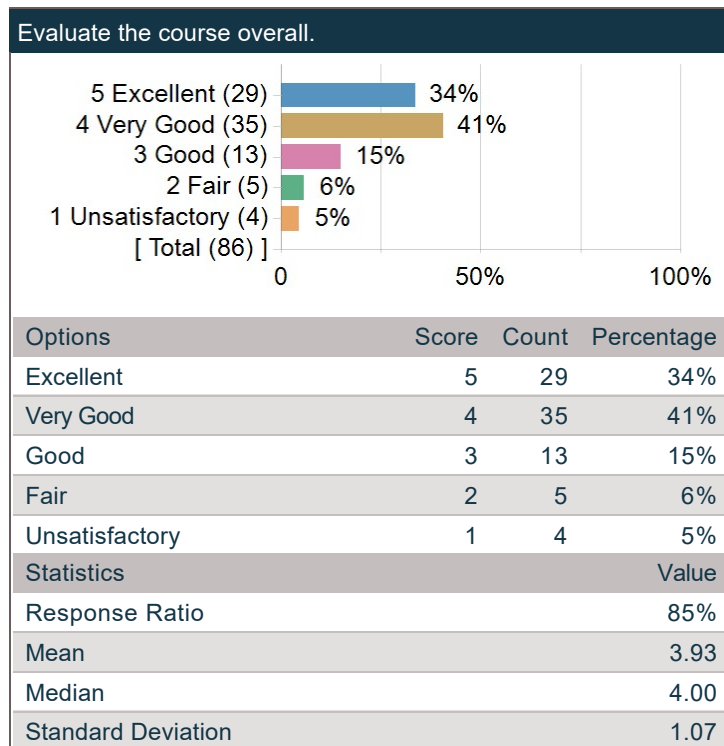
Creation Date: **Tuesday, December 20, 2022**

## General Course Questions

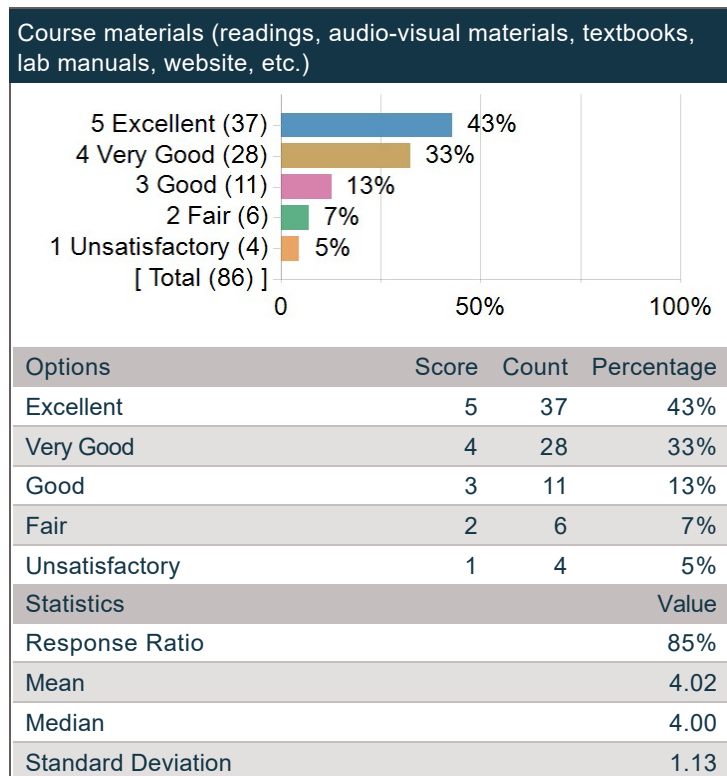
### Course General Questions

	Count	Excellent	Very Good	Good	Fair	Unsatisfactory	Course Mean	Dept Mean	Division Mean
Evaluate the course overall.	86	34%	41%	15%	6%	5%	3.93	4.08	3.99
Course materials (readings, audio-visual materials, textbooks, lab manuals, website, etc.)	86	43%	33%	13%	7%	5%	4.02	4.18	4.05
Assignments (exams, essays, problem sets, language homework, etc.)	86	34%	37%	16%	8%	5%	3.87	3.84	3.83
Feedback you received on work you produced in this course	85	42%	29%	18%	7%	4%	4.00	3.80	3.70
Section component of the course	70	40%	34%	17%	4%	4%	4.01	3.85	3.86

### Evaluate the course overall.



**Course materials (readings, audio-visual materials, textbooks, lab manuals, website, etc.)**

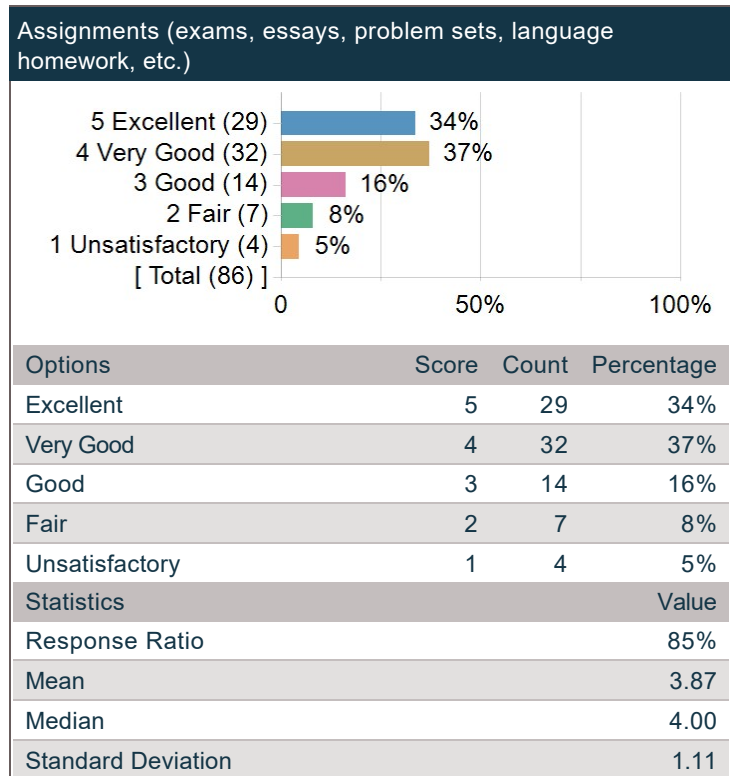


**Add comments about course materials?**

Comments
The textbook was very clear and comprehensive. The recording of lectures was also helpful.
As someone with no theoretical background coming in, I got lost easily in lecture and often could not understand the textbook. I had to rely on lots of youtube videos to really understand topics, and ask multiple times in office hours.
The textbook is very good and comprehensive, but there are mistakes throughout.
The textbook was very helpful with many of the concepts in the course.
Lecture and section was good but solutions released for psets and Section were often not helpful and did not fully explain the problem.
I wish that searching the textbook for propositions and keywords was easier.
The textbook was well organized and fairly easy to follow, though it did have some mistakes. I'm glad we were able to fix these ourselves via pull requests – thank you for making my urge to correct every minor error productive!
Lectures felt a bit rushed sometimes, but the rest of the materials were excellent.
The textbook was engaging to read, and section materials were helpful. Nit: I wish section notes and problems were put together as one PDF.
The textbook is a great resource.
Book is solid but at times can get confusing / ambiguous
Boaz's textbook is great!
The course has amazing content, really exciting and useful.
Textbook was very helpful to read through, some parts were a bit confusing but Perusall was instructive for asking questions and clearing up confusion. Also, course staff was very responsible on Ed. Ed was very disorganized — it was a nightmare to find problem set solution/section handouts/solutions, and often there were 10–20 posts pinned at the top (which totally ruins the point of pinning a post). It would be helpful if there was one centralized post for problem set releases, one centralized post for solutions, etc. that were updated each week and kept pinned at the top of the Ed.
What was incredibly notable was this course's use of Perusall for the textbook, and the fact that the professor himself would respond to any questions asked within a few hours.
The textbook was very good and very helpful in learning the materials.
Textbook was very dense.

Comments
Textbook was very helpful for the class. Section notes were fine.
The textbook can be a little tricky to read at times.
The textbook is unnecessarily verbose. It's hard to find the important theorems.
Very well written textbook, really helps a lot in conjunction with lecture and section materials
There is a lot of material in the book/readings that are not even needed to do well in the course. As a result, fully diving into readings often felt unnecessary.
The textbook is long-winded, contains unclear examples, and has many errors.
Very good textbook, though there's still room for improvement – utilizing students' help to improve it was a very good idea
Organize them better!!!! So frustrating to search through Ed
I think lectures and textbook complimented each other well, I didn't enjoy Perusall as a platform but of course there was flexibility with that
There was a lot of dense reading assigned, which made it difficult to get through.
The slides were A+ at highlighting core concepts from the textbook! I also appreciated being able to ask questions on specific textbook sections via Perusall, something I was never able to do in a course. However, I ended up forgetting to cursorily scrolling through it, because when I do my readings, I use the online/PDF version so it's easier to copy & paste, & Zoom. If possible, perhaps there are no explicit Perusall assignments, but participation on it will still be looked at?
So much available, lets students find what works for them
The book is incredibly helpful and very well-written. The discussion platform Ed is also very helpful for clarifications and asking questions! Perusall is also a super helpful platform to understand challenging concepts!
Great book! I'm not a fan of Perusall. (Tt feels cleaner to have a pdf on my desktop, and I want to read in private).
The textbook was fantastic. The only reason I don't give it an 'excellent' is because it was quite difficult to search for terms or specific theorems (Command F didn't work very well for such things as P/poly, for instance). If there were a glossary of terms/theorems, I think that would be hugely helpful to future students.
The course materials were well-organized, but not particularly engaging.
N/A
Overall, I think the class materials were fair and broad.
I found the textbook a little difficult to use at times. One pain point I had was the difficulty in searching for particular subjects when I didn't know what chapter it was in. I also think it would be helpful if some kind of summary sheet was provided that included all of the crucial ideas and theorems we should know.
I don't really like Perusall as a platform, but the book is well written.

**Assignments (exams, essays, problem sets, language homework, etc.)**

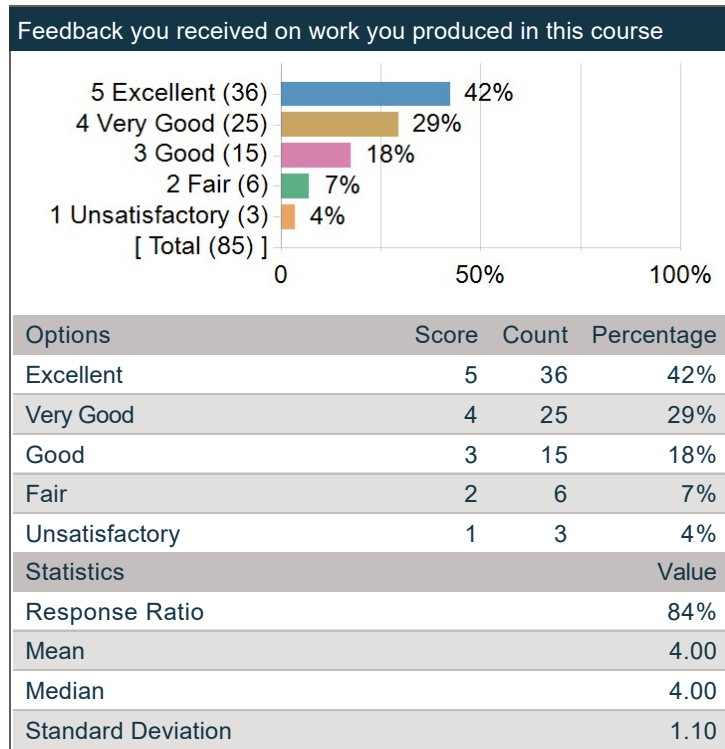


**Add comments about course assignments?**

Comments
The assignments targeted the material well and the exams were comprehensive.
I thought that the psets were hard but doable with office hours. Bonus points made them less stressful.
The exams and problem sets have all been very fair, with opportunities to gain bonus points so that even if you miss some points you can receive a very good grade.
I think the Psets were honestly very well written, albeit a little repetitive sometimes.
The exams were never too difficult.
LatTeX files for problem sets were very well organized. The bonus problems are great, although sometimes quite difficult!
Sometimes unnecessarily long in ways that did more to keep you busy than challenge your thinking, but overall good assignments I'd say.
Problem sets were fascinating, helpful, and relevant to course concepts. The reflections were also a great way to gain more thorough understanding of how to apply (not only internalize) the material.
The homeworks were very good. They helped increase my understanding a lot!
Some of the problem sets felt cumbersome and not very rewarding.
This course is incredibly challenging and very time-consuming. There were especially a few times when it was very difficult to delegate time between studying for exams and working on problem sets. Even in normal weeks outside of exam periods, there was often a need to make a trade-off between completing the problem set versus reading the textbook thoroughly versus watching lectures on time.
The problem sets were usually very helpful, but were often very difficult and time consuming.
Problem sets were usually a reasonable amount of work (the DNA computer seemed a little over the top). The midterm averages were surprisingly high. Either they should have been a little harder, or Boaz did a very good job.
Problem sets were really interesting!
Some pset problems were confusing and unnecessarily hard.
Problem sets helped with learning the material but some were much longer than others, especially towards the middle of the semester
Problem sets often had questions that did not really test knowledge of material or involved overly difficult questions that went outside the intended scope of the course.

Comments
The bonus questions were pretty fun, but some of the questions were truly too tedious (like DNA computer).
The problem sets were focused on grading, not understanding.
Some of the problem sets felt unrewarding / unnecessarily difficult (e.g. DNA Machine) – though the concepts themselves were cool
I think maybe one less Pset would have been more manageable but loved the option for bonus points!
Problem sets could be quite challenging but a combination of great staff in office hours and the relief on grades from revisions and bonus questions made them overall positive!
Boring but easy
Some psets were much longer than others. Also the expected lengths of the responses were unclear. Sometimes the staff solution would be extremely short when my write up was over a page long.
As stated before assignments were really long at times
I thought problem sets were pretty well scoped!
The opportunities for bonus and to complete your understanding using the course assignments is wonderful. It promotes students to understand the material, rather than simply finish it!
I like that the psets are easy and with a lot of extra credit. It makes this class non-stressful.
I really appreciated the bonus points and reflections, which took some of the stress off of completing problem sets. Thank you for that!
I often found the problem sets to be overly challenging, and it would eat up the entirety of my Monday, Tuesday, Wednesday, and Thursday night.
Problem sets were useful, but varied in length and time expenditure by a significant amount, which isn't very fair to students. Offering a chance at reflection on psets was very helpful. For exams, TFs should be spoken to about how problems should be graded. I felt that some TFs graded rather harshly on some problems and offered no partial credit, whereas others did offer partial credit, and the lack of partial credit that I was normally accustomed to reduced my morale and the morale of others I've spoken to as well. I appreciated the multiple chances we had at extra credit because it helped me review and reflect on the learned material.
N/A
I think the psets help you learn the material better, and exams were very fair. Exams included a lot of bonus points, so this reduced the stress.
I think all of the problem sets are a fair length and a good difficulty. I also really appreciated all of the extra credit that was offered.
Psets were pretty good, with lots of opportunities for extra credit, which is always nice.

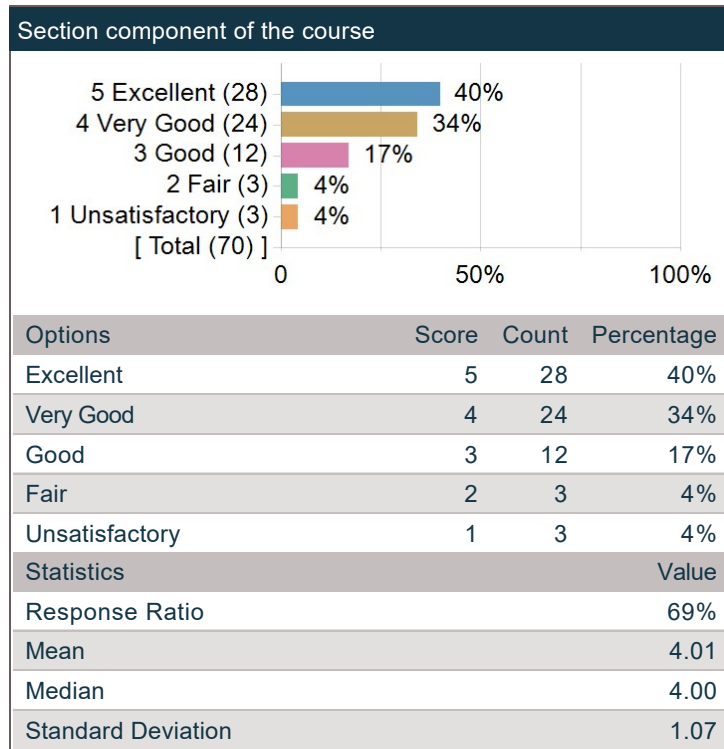
### Feedback you received on work you produced in this course



### Add comments about course feedback?

Comments
All feedback has been very good and constructive.
Whenever I got some question wrong there was always good feedback and graders were responsive for regrade requests?
Fast, quick, generally accurate.
As per the previous course improvement comment, some of the grading felt extremely nitpicky where graders took off a lot of points for small details even though the solution was very reasonable and demonstrated an understanding of the course concepts.
The feedback in this course was good.
Gradescope comments were usually relatively helpful.
Graders were extremely unreasonable and stubborn with regrade requests.
Problem sets and exams were graded fairly
The TAs that provided feedback on homework were often lazy and careless. They often made mistakes when grading and provided unsatisfactory solution sets. They took off unreasonable amounts of points and made no effort to understand the effort that a student made to comprehend a problem and were only focused on results, not process. Feedback was often published at a time that was later than ideal and provided no enhancement to student growth.
No issues here
Usually very clear and detailed, I learned a lot and the revisions were helpful to revisit this as well
Grading was arbitrary. I lost a lot of points because the grader's missed the main idea of the proof.
Gradescope comments were helpful :)
Thorough and correct.
It would have been nice if the solutions to the second midterm were released. Additionally, it was at times unclear to me from the Gradescope comments why I lost points on a problem.
N/A
I think feedback was mostly fair. Sometimes, the feedback on grading was very dismissive and basically deducted points without explaining anything.
We received grades very promptly, which was very handy.

**Section component of the course**



**Add comments about the course section?**

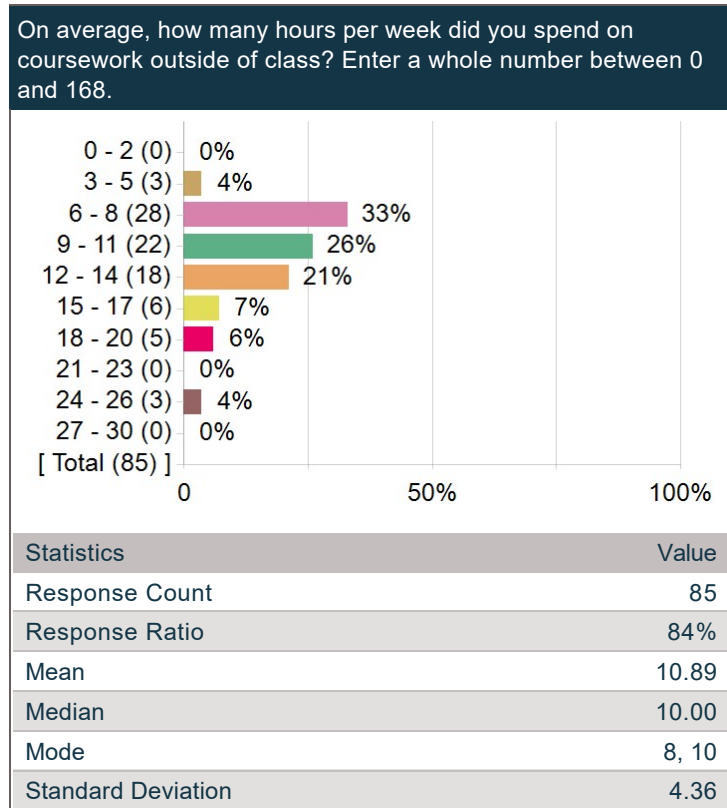
Comments
Sections were very accessible.
Loved section — best part of the course.
Section quality seemed to vary, with some section leaders teaching directly from their provided section notes and some going more above and beyond.
Section schedules were somewhat difficult to make work at times.
The sections were very useful, and helped greatly in learning the material.
As mentioned previously, it just felt like I was hearing the same information for the third time (textbook, then lecture, then section). The practice problems were probably one of the most useful parts, but we only had a few questions each week.
Tfs are great!
Section/office hours was the most helpful part of the course.
The TFs in this course were amazing and the sections were instructive and enjoyable
Depends on section leaders but could be very helpful!
I enjoyed the balance of concept review with some practice problems :)
Section was helpful and well-run. CS 121.5 was pretty great sometimes. Not all of the lectures were good, but some were, and it had a bit of community. (I bet it could have more.)
Section had good materials and was well-taught.
N/A
Section was a good overview of the lectures, and it was especially good for simultaneously enrolled students who cannot attend physical lectures. Sometimes, the problems were a bit repetitive though.
I never attended.



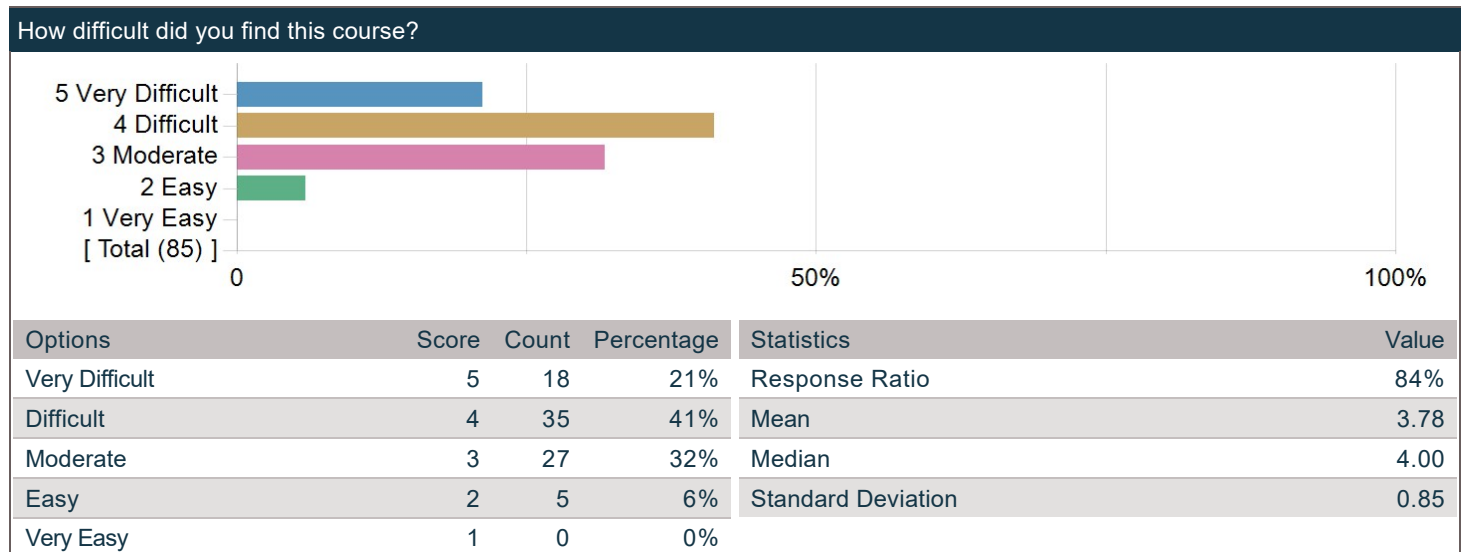
## Requirements - What did this course require of you?

On average, how many hours per week did you spend on coursework outside of class? Enter a whole number between 0 and 168.

Frequency chart and mean excludes students who answered 31 or more hours.



## How difficult did you find this course?

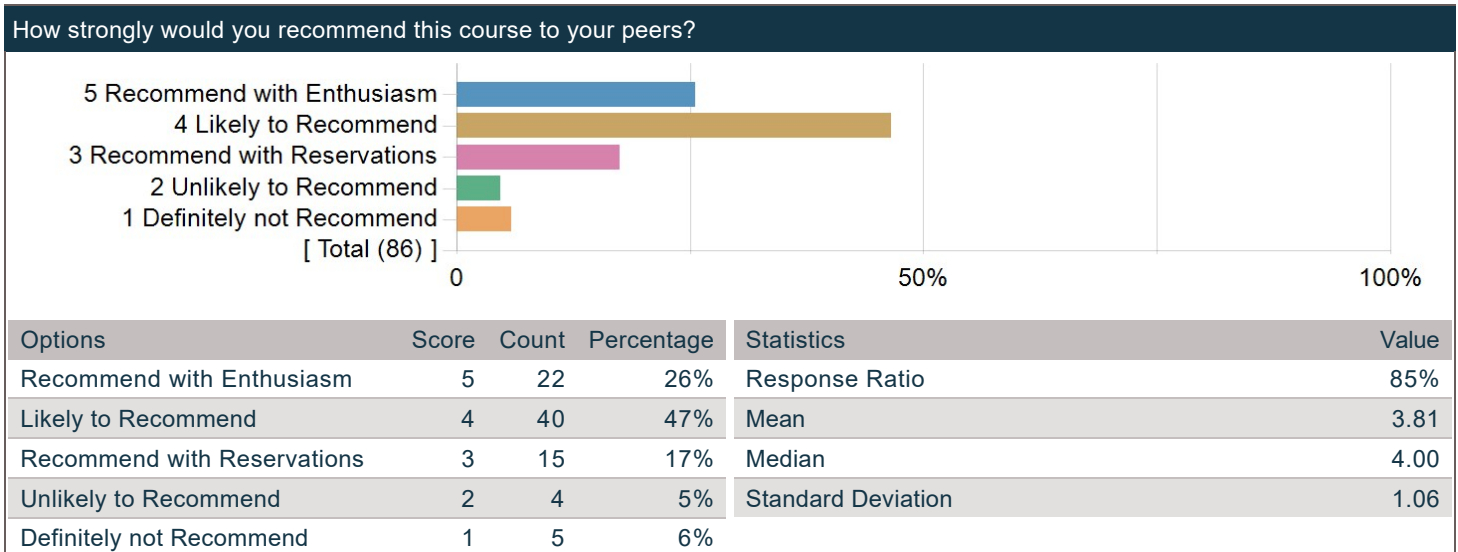


**What was/were your reason(s) for enrolling in this course? (Please check all that apply)**

Options	Count
Elective	5
Concentration or Department Requirement	82
Secondary Field or Language Citation Requirement	1
Undergraduate General Education Requirement	0
Expository Writing Requirement	0
Foreign Language Requirement	0
Pre-Med Requirement	0
Divisional Distribution Requirement	0
Quantitative Reasoning with Data Requirement	0

**Recommendations - Would you recommend this course?**

**How strongly would you recommend this course to your peers?**

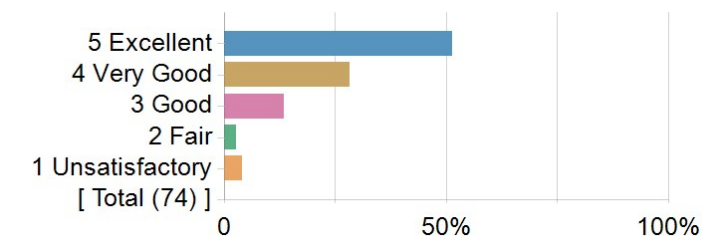
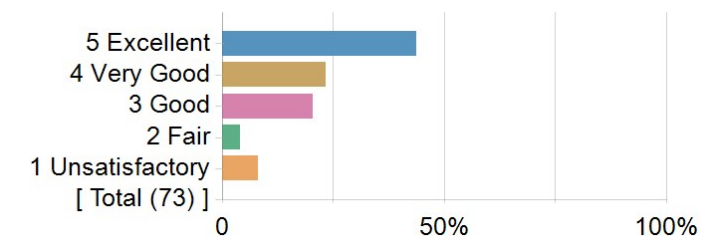
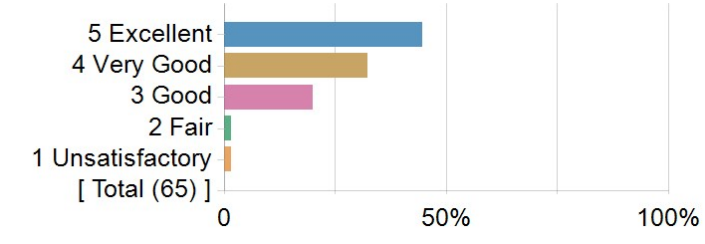
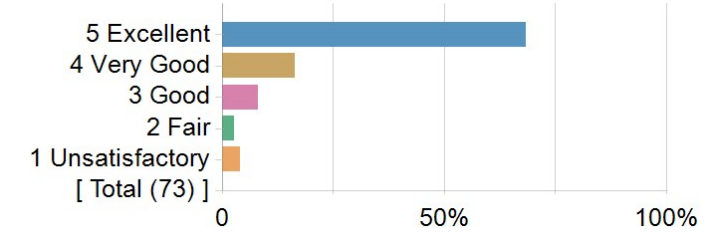


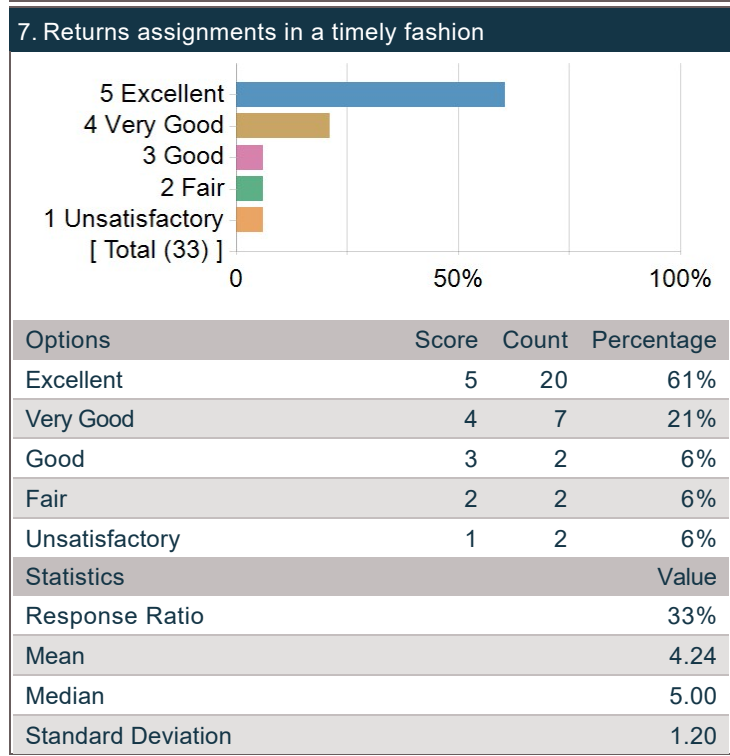
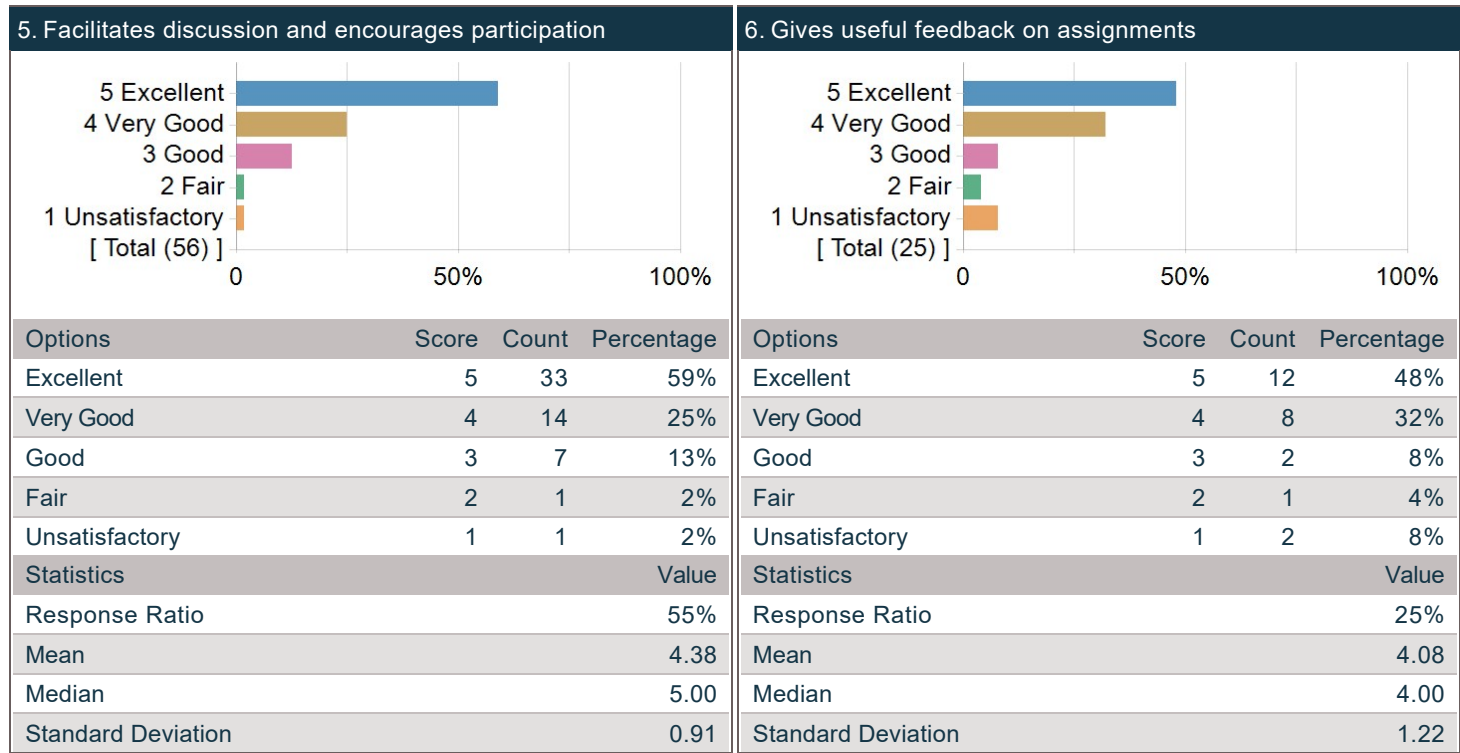
**Evaluation of Instructors**

**General Instructor Questions**

	Count	Excellent	Very Good	Good	Fair	Unsatisfactory	Instructor Mean	Dept Mean	Division Mean
Evaluate your Instructor overall.	74	51%	28%	14%	3%	4%	4.20	4.39	4.31
Gives effective lectures or presentations, if applicable	73	44%	23%	21%	4%	8%	3.90	4.27	4.19
Is accessible outside of class (including after class, office hours, e-mail, etc.)	65	45%	32%	20%	2%	2%	4.17	4.14	4.20
Generates enthusiasm for the subject matter	73	68%	16%	8%	3%	4%	4.42	4.48	4.40
Facilitates discussion and encourages participation	56	59%	25%	13%	2%	2%	4.38	4.40	4.32
Gives useful feedback on assignments	25	48%	32%	8%	4%	8%	4.08	4.26	4.12
Returns assignments in a timely fashion	33	61%	21%	6%	6%	6%	4.24	4.23	4.05

**Instructor**

1. Evaluate your Instructor overall.				2. Gives effective lectures or presentations, if applicable			
							
Options	Score	Count	Percentage	Options	Score	Count	Percentage
Excellent	5	38	51%	Excellent	5	32	44%
Very Good	4	21	28%	Very Good	4	17	23%
Good	3	10	14%	Good	3	15	21%
Fair	2	2	3%	Fair	2	3	4%
Unsatisfactory	1	3	4%	Unsatisfactory	1	6	8%
Statistics			Value	Statistics			Value
Response Ratio			73%	Response Ratio			72%
Mean			4.20	Mean			3.90
Median			5.00	Median			4.00
Standard Deviation			1.05	Standard Deviation			1.25
3. Is accessible outside of class (including after class, office hours, e-mail, etc.)				4. Generates enthusiasm for the subject matter			
							
Options	Score	Count	Percentage	Options	Score	Count	Percentage
Excellent	5	29	45%	Excellent	5	50	68%
Very Good	4	21	32%	Very Good	4	12	16%
Good	3	13	20%	Good	3	6	8%
Fair	2	1	2%	Fair	2	2	3%
Unsatisfactory	1	1	2%	Unsatisfactory	1	3	4%
Statistics			Value	Statistics			Value
Response Ratio			64%	Response Ratio			72%
Mean			4.17	Mean			4.42
Median			4.00	Median			5.00
Standard Deviation			0.91	Standard Deviation			1.04



## General Course Questions - Comments

**What were the strengths of this course? Please be specific and use concrete examples where possible.**

Comments
The homework assignments were reasonable and helped students better understand the material. The homework policy was also forgiving with the bonuses, which I think encouraged students to focus more on learning than grades.
1) The teaching staff — some of the TFs are incredible and help a lot. 2) Lots of opportunity for bonus points — made the psets much less stressful 3) Variety of learning resources — textbook, lectures, section notes, etc — easy to find what works best
The course content was fascinating, and course support in the form of office hours and section meetings was very strong.
We cover a lot in this course, and there are many opportunities to raise your grade with bonus points.
This course was a great introduction to theoretical computer science, and I think the material was very well-organized. The Psets also contained many great example problems of the types of problems we would encounter on tests, and they always followed very closely with what we did in class that week.
The textbook is great. The lectures are fun. I especially loved the "adversary" for the pumping lemma. It just made using the pumping lemma click! Whenever I see quantifiers, I think of the adversary analogy now.
Interesting material, problem sets were very well-crafted, and the copious amounts of extra credit reduced grade pressure. Prof. Barak is a great professor.
The book is excellent and very comprehensive. I found the material inherently interesting. Lecture often helped clarify and solidify big-picture take aways.
Prof. Barak taught passionately, and the teaching staff devoted time and effort into this course and its students. The material was engaging and genuinely fascinating, and the problem sets were rewarding to work on and use to better understand the material.
– Strong course staff and a lot of infrastructure – Homeworks that were not overly stressful but really helped develop understanding of material! (x2'ing this—they were really good) – Generally fair exams
The material is very interesting.
Some concepts are really cool (halting problem, godels) and you get to learn about them
The material in the course is extremely interesting and it is covered with a very accessible pacing. The course had an extremely supportive course staff.
Amazing content! Well organized!
Good course content on theoretical CS. Personally, I found the pset reflections to be helpful. In other courses, I tend to just skim over my pset when it is returned, but since I had to do the reflection for this course, I was able to more concretely think about what I had gotten wrong and learn from these mistakes. Additionally, I appreciate the multiple opportunities for bonus points in the course since I could focus more on learning the content rather than trying to get perfect grades.
This course is incredibly well-taught and well-supported, with several different resources available — from a textbook made for this course specifically and detailed sections to several practice exams and an incredible staff. The CS121 staff is honestly unlike any other, and what was especially notable was the professor's dedication to the course. Help is always there when you need it.
Good lectures and textbook, very compelling material.
The lectures proceed in a logical sequence; the book is pretty helpful!; the psets are challenging but reasonable if you go to office hours; the exams are generally reasonable and support resources are helpful.
A new perspective into the theoretical aspect of computer science.
The course had well-organized materials, reasonable and interesting assignments, and great feedback from TFs.
There were plenty of office hours and good extra credit opportunities (i.e. pset reflections and a bonus pset).
+ Lecture content and textbook line up extremely well. + Problem sets were actually reasonably interesting. + Lots of opportunities for extra credit. I was never very concerned about my HW grade.
The textbook was very useful and informative. The teaching staff went above and beyond to make office hours and section available to students.
Probably one of my favorite courses that I've taken at Harvard so far. Boaz clearly really cares about the content he teaches and about his students. I thought that the lectures were clear, problem sets were interesting and helpful, exams were (more than) fair, and sections were great (shoutout April and Priya).
CS 121 is an excellent class that delves into a lot of different topics. While more theoretical subjects are often under appreciated, Boaz does an excellent job of helping students understand the beauty and importance of many of these concepts.

Comments
Boaz is such an incredible lecturer. Additionally, I love how he structured the PSET grading system
The course was very well designed and the different components (lecture, readings, sections and problem sets) complemented each other very well, helping with the process of gradually learning the material rather than having to study it in a concentrated manner prior to exams.
The course provided an in-depth exploration into theoretical CS. The course was very organized and well-structured, with clear themes and "big ideas" that were elucidated upon in a straightforward way. I enjoyed the tree of topics which gave me a clear idea on where we were in the course. The readings served as a great preparation for the lecture, which reinforced the material. Finally, the pssets provided hands-on practice which really solidified the material without adding extra confusion. Course communication and TF/instructor response time was fantastic. Peer and office hour support was plentiful and excellent. For the exams, plenty of practice material was given, so I felt that I knew what to expect coming into exams, which I appreciate a lot.
The strengths of this course were the teaching fellows leading section and office hours. The manner in which they explained concepts was very clear and useful. Section materials were also very helpful.
The book is very well written, and the course material is laid out in a logical manner. The assignments are generally reasonable, and the course staff is very accessible and helpful, with lots of office hours and sections.
This course has good material and covers a satisfactory range of topics in a reasonable amount of time.
The content, teaching staff, textbook
The material is really interesting and the PSETs and Office hours are good and textbook is decent
I learned a lot between the staff and Professor Boaz
Material is interesting
I think there were several aspects of the course that reduced stress. The bonus points on pssets and exams in particular were great.
Theoretical computer science is an interesting field and this course exposed me to it. Several pset problems were particularly engaging. The teaching staff is highly accessible and helpful.
I like the fact that we have a book accompanying the lectures. Lectures slides were also nicely prepared with different colors etc. Exams seemed fair and I think it was reasonable to not curve the class since I think there is huge selection bias for the people that self-select into CS121 instead of CS120 which leads to a very skewed curve.
I mostly appreciated the course structure that emphasized learning over outcomes as much as possible — in particular, the problem set reflections actually encouraged me to reinforce and think more slowly over former concepts, and helped contribute to studying for midterms/finals! I also appreciate the scaffolding for less/more comfy students, such as with the bonus questions.
Organized material, rigorous thinking process and math
Great overview of many topics in theoretical CS, lots of teaching support and extra credit opportunities.
None
Eloquent connections to the limits of computation and philosophy!
The course material is incredibly interesting! The textbook is fantastic, and all of the lectures are exciting and organized! The TFs are wonderful, and the opportunities to improve your understandings are phenomenal.
The course builds the foundations for abstract computer science, building from NAND-circuits to Turing machines and quantum computing, providing a high-level introduction to a ton of concepts.
The content in the first half was some of the coolest stuff I've ever seen. The textbook's a great reference. There are plenty of sections and office hours. There's lots of extra credit, which means if you take the time, you can do very well.
There is a ton of support for the problem sets on Ed and during office hours. The problem sets were generally interesting, but quite challenging.
Well-designed assignments that made me think about the course material.
I really enjoyed Boaz's teaching and think the class is well-structured in terms of office hours, the textbook, and problem sets. It feels very streamlined and well-run.
Second half is interesting, first half is garbage.
Important and interesting course material. Textbook was a useful read. Section materials were helpful too. Elijah, Patel Fellow, was a savior.
Interesting material
textbook was phenomenal, pssets were very fair
I think the class gave a good overview of the classical areas in theoretical computer science.
The bonus points on the PSETs really encourage students to try to solve the problems on their own.
broadened my understanding of theoretical computer science
Covers a lot of core ideas about computation theory. I also appreciated how thoroughly we went through each subject, and I think I

Comments
have a very solid understanding of all the major concepts now.
Boaz is an engaging lecturer and really knows the material.
The teaching staff was very helpful. The content was interesting

**How could this course be improved? Please use concrete examples where possible and provide constructive suggestions.**

Comments
1) Grading on the exams was brutal — I sometimes felt like I didn't get any partial credit for questions I did more than 50% of the work for. Made me feel very helpless. 2) More representative resources to prep for midterms. I felt that the practice exams were nothing like our exams. Some of the practice material didn't have solutions. 3) Lectures and the textbook were both difficult for me to process. I relied more on section/office hours, but I felt that section notes were not comprehensive at all.
Boaz's lectures often felt hectic and all over the place, with mistakes within his slides that he corrected verbally but should have caught earlier to avoid confusion. Also, I am never a fan of receiving grading based on ed / perusall participation. I feel that it promotes useless comments and questions that bog down the forum, as people are encouraged to engage with these platforms as much as possible, leading to many comments on ed posts that are redundant or simply unnecessary. Also, I feel as though it should be the TFs' or professor's role to answer students' questions and fix mistakes in official course materials, not the students who are still learning the material.
The median score on the second midterm was 103% (one of many crazy examples), which personally made it very hard for me to tell where I stood grade-wise in the class. I honestly wish the bonus was tuned down just a little so that grades could more accurately reflect achievement in the class.
A few of the pset problems were a pain. For example, the dna tm was yeah... The enormous amount of available bonus points helped a lot with these since I felt comfortable trying something and not stressing too much.
Attendance/participation expectations were not made clear enough.
There was a bit too much time spent on reductions in my opinion. I understand that it's very fundamental to theoretical computer science, but it started feeling like busy-work to type them up again and again on the problems sets. Perhaps add some questions that only ask to come up with the idea for reduction without spelling out soundness and completeness in full detail, otherwise it just becomes a drag to type them all up.
I wish the course's Ed were more organized, but other than that, the course was very well-taught and organized.
– The textbook felt unnecessarily verbose in places. I spent a lot of time making sure I read through the details, for instance, of the universal TM, when it was not necessary. – Participation paternalism? I really wasn't sure how I was doing in "participation" the whole time as a simultaneously enrolled student; even though I was up to speed on material, I didn't like that the class policed my attendance. However, it did force me to go to section + watch lectures on time, a good thing.
I thought the lectures could have been a bit better structured, as well as the class as a whole. We spend a lot of time just discussing turing equivalences, which I found more cumbersome than insightful. The class starts to pick up in the second half, but the first half seems largely wasted and could be shortened considerably. The problem sets were also a bit too easy in my opinion; it would have been nice to see more questions that required more than just applying the exact ideas discussed in class, and provided a deeper insight instead.
Make the material less dense, or put an advisory warning to students to not take this class if they're not Math 25 / Math 55
To me, the grading in this course didn't feel like it tested for understanding, but rather nitpicking for if we said the exact key words on the rubric in our solutions. I think in many instances, it was clear that I understood the problem and wrote out a reasonable response, but then I got harshly graded because I didn't say things in the exact way the grader was looking for. Additionally, sometimes I would cite things that we had already covered extensively in previous courses I had taken (like CS124), but then I would get penalized in CS121 for lack of explanation. It would be a better experience if the course staff were more aware and considerate of the breadth of previous preparation that students have coming into CS121, and graded for understanding rather than docking off lots of points for small details.
The lecture material was incredibly repetitive to the textbook material, and it was challenging to manage time between the two since the textbook is quite hefty. I wish there had been a better system of knowing what to focus on in the textbook, or maybe some diversification between the textbook and lecture.
The lectures can be unfocused; a lot of notation or convention seems arbitrary and confusing (lots of constants = 10); the bonus point system is strange and makes grading unpredictable; required usage of Perusall can be a turn-off.
A lot more structure and closely related practice materials to problem sets, and more definitional practice.
I wonder if the lectures at the end of the course could be more related to each other—I don't feel like I retained all that much from the

Comments
quantum and ML lectures.
The course was extremely stressful and time-consuming. I've gathered from many students in the course that it was the #1 cause of stress and low amounts of sleep this semester. Psets felt designed to be overly-difficult, and would practically require attending office hours for any chance of success. Although there were plentiful support resources in this course, it was very difficult for me to find success in this course as a student who switched into the Computer Science concentration last year, despite having taken CS20. I hope the course staff considers shortening problem sets, simplifying questions such that most of the problem set can be completed independently, and weighing them more heavily compared to midterm exams in future versions of this course.
<ul style="list-style-type: none"> <li>– Section didn't always feel as helpful as they could have been. It mostly seemed like the section notes repeated what had already been said in class, usually in a similar manner.</li> <li>– Capping the HW grade at 100% disincentives doing work towards the end of the semester.</li> </ul>
Lectures were sometimes unclear and did not always feel worthwhile as the same content was often available in the textbook.
It's hard at times to read a textbook that's still in-the-works. There were times when typos impeded my understanding, and some proofs were a little overly dense. I also wish we had covered space complexity – or at least touched on it.
I wish there were more Zoom sections
The course moved a bit more slowly than I would've liked. I found section material to be quite repetitive and overkill. There wasn't much incentive to do reflections or bonus problems.
The course textbook is overly convoluted; it involves too much material which is not even tested on exams or mentioned in problem sets. In addition, lecture was not as engaging as it could have been and certain key concepts were not detailed enough in lecture.
It was a little bit redundant to read both the book and watch lectures sometimes, and I felt that it would be interesting to see some slightly different material. I also think that we could have moved through the material a little bit faster and covered some more of the interesting specifics, especially during the NP section.
This course would be improved if it was taught by a professor who cares less about grades and more about students learning the material in whatever way works best for them.
More live problem solving, better website? – or at least a place where students could quickly refer to section notes / solutions (even a Google Sheet), since searching for them through all the Ed posts was a major annoyance
The organization is really really bad. Why are all of the course materials posted on ED? it should be a more centralized and more easily accessed system to get section notes, lecture notes and psets etc. Every other class I have ever taken has had a much better system for this, by far the worst course out of my 4 years at Harvard in that respect. The lectures aren't the best explained, I think Boaz needs to be better at teaching this course to be frank
Overall can't think of too much
The textbook and lectures terse and unreadable. The proofs are impossible to parse and don't emphasize the main ideas. Boaz's writing style is impossible.
I think this course was a decent bit of work, and the psets were not consistent in their length and the amount of time it took to complete them.
There is a lot packed into this class without clear reasoning as to why, which is often frustrating given the difficulty of the material. Weekly psets make it very difficult to take this class along other time-intensive classes. The post-midterm 2 section seemed to consist of Boaz discussing his other CS interests in tangential manner to theoretical CS, and I did not feel that we learned these topics with enough depth to appreciate them. Overall, I fear that the plethora of details crammed into the course sometimes obscures the bigger picture.
Psets sometimes felt extremely cumbersome to type up to the point where I felt like I spent 1/4 thinking about problems and 3/4 writing boilerplate latex etc.. I think I could have been a bit more lenient since we had extra points to make up for any small mistakes/incoherences but I didn't know that we would have so many bonus points at the end.
For the exams, I am a bit torn between extra credit: During my section the TFs recommended basically not even attempting the EC (since with Hesterberg EC was 0 points and way harder than the rest of the exam) so I did not even look at it during Midterm 1. Well after results came out, I was below the median – I found out that most people (at least the ones I knew) did EC and it was not that hard at all. So I think EC is good (when capping each component to 100%) since it takes of the pressure to do perfectly fine, but please communicate the fact that EC is in fact very approachable and can help you to make up for questions you missed / did not know.
While Ed is a great discussion board, it's not a database for class material. It was really cumbersome to find materials scattered across Ed (especially since search is so bad on Ed).
I wish there were a few review sessions over reading period! I also was unsure of expectations for thoroughness in problem sets; I felt like my proofs were overkill sometimes since I took CS124 previously and erred on proving *everything* when it perhaps wasn't most effective for the course material in CS121.



Comments
It also was hard to find some files/posts on Ed sometimes, and they wouldn't always be updated on the website.
No evening exams on school holidays in the SEC please! And even though it was great to cover lots of topics, it sometimes felt disjoint / unrelated (especially the ML, quantum, etc chapters at the end)
In every possible way: teaching, psets, section, and the rest.
Maybe add an anonymous online Q&A website like the one from CS61.
I would love for the more-comfortable sections to be recorded if students can't make it to that time!
Some of the practice exams didn't have answer keys.
Even more class interaction during lecture. I always like when a lecturer asks the class questions (like, here's a problem to talk about, or questions that lead us to discover something).
Some of the lectures were rushed, especially the topic lectures at the end.
Though Boaz obviously put a ton of effort into the class, I didn't think he was the best lecturer; the one lecture that Madhu gave made me realize how good a lecturer could be. The flow of information was difficult to follow during Boaz's lectures; at times, I felt like I didn't gain very much from attending/watching the lectures.
I found the material to be disengaging and extremely challenging, which is not a weakness of CS121 itself, but just an inherent reflection of the material. I wish the theory requirement was not as intense for the CS concentration at Harvard.
First half of the course is confusing and unnecessary. The syntax is all made up and will never be useful outside of this class.
TFs that I had were not the most helpful in OH and did not guide students to the solution if they were struggling to understand the materials. Lecture was in general not incredibly clearly outlined or organized before the middle of the semester.
Better lectures
lectures were very poor and unorganized. participation component of grade is unnecessary
Even though the textbook has a really good motivation and definitely on the way to be a great textbook, I think it is sometimes not very clear. I especially think this was true of the first chapters.
a little less dependence on the text book, since the text seemed very dense to me
I would have enjoyed spending more time on more recent applications such as the few lectures on machine learning and quantum computing at the end of the semester.
I'm not a huge fan of the flipped-classroom style where readings are due on Perusall before lecture; I'd generally much prefer to just be introduced to the material in lecture, then be given supplementary material about proofs if necessary.
Lectures may be a little haphazard.

## Requirements Comments - What did this course require of you?

### In your opinion, what preparation or background is necessary to take this course?

Comments
An adequate background in mathematics is helpful. I also think prior knowledge of probability and statistics would be helpful for the end of the course.
Maybe some proof background/stats/math
Some programming experience and substantial experience with writing proofs.
Knowledge of proofs at the CS 20 level
In my opinion, background in computer science is not necessary, but knowledge of some discrete math would help a lot. This is really more of a math-heavy class than a CS-heavy class, so an extensive math background can also make a big difference. The public Homework 0 on the course website is a pretty good indicator of what the course is like.
Mathematical Maturity
Proof-writing skills (CS20 or similar should be enough)
Some discrete math perhaps, but you really do learn most of what you need to know.
Mathematical maturity.
CS20/120 or equivalent familiarity with discrete math, some mathematical maturity.
Math 25/55, Compsci 20
A good understanding of a few topics from discrete math and a relatively good reasoning ability.

Comments
Strong proof background beyond the level of CS20
Mathematical proofs
Programming experience (CS50/61). It's helpful to have math/proof experience (CS20, for example) and some algorithms experience is helpful, as well.
Very solid math and proofs background and familiarity with the fundamentals of computer science.
I would recommend a gentler intro like CS 120.
Discrete mathematics.
Ability to write proofs and basic programming skills are essential.
CS20 and CS50 at the very least. Other proof-based math courses would have been very helpful.
I took CS 124 before taking CS 121. I'm not at all trying to say that should be a prerequisite, just take my opinion with a grain of salt because I probably came in with far more familiarity with theoretical CS than the average student. Experience with writing with proofs is a necessity. Familiarity with discrete mathematics is also a must. The amount you need for this class is not too large, but you should at least be familiar with big O notation.
Proof background at the level of math 22+ or CS 20 or both
I can't imagine doing this course without some kind of exposure to proofs – whether that's Math 22/25/55 or CS 20. Being comfortable with some probability at the level of Stat 110 is super useful for probabilistic computation.
A background with proofs would be helpful.
CS 20 or equivalent
Some proof-based math
CS20 or equivalent preparation
No background is necessary! Comfort with graphs and mathematical notation is nice.
CS50, CS20, and CS51/61. Practice writing proofs is very helpful.
Not much, but a solid grasp of mathematical reasoning is very useful to have.
Discrete mathematics
Having a background in discrete math / proofs is useful but not necessary. CS20 / Math 22a and above should be plenty of preparation
Mathematically mature in proofs
Definitely as much proof based math as you can have, I think Stat 110 was somewhat helpful and CS120 was helpful too
Mathematical maturity
CS 50, Stat 110
Familiarity with CS — this would be a hard first CS course. Stat 110 or probability intuition. The more CS background and particularly CS course at Harvard background you have, the better prepared you'll likely feel
Solid proof wiring.
CS20 or equivalent discrete math experience. I think it was tremendously helpful to take CS124 before, although conversely it would be tremendously helpful to have CS121 before CS124. I do think CS124 was more challenging, so CS124 to CS121 definitely prepared me well!
Rigorous proof, discrete math
Some theory and proof writing
Lots of math and little to no coding required. I'd put this course in the math section for those reasons.
Previous proof experience.
Proofs and knowledge of general concepts in CS, e.g. graph theory, circuits, etc.
Not a lot is NECESSARY, but being good with math and proofs is pretty helpful.
Discrete math knowledge, comfort with proofs.
Taking COMPSCI 20 is probably a good level of preparation
CS 20
Good studying skills
Previous experience with computational theory is helpful, as well as some statistics/math background.
Math22a

Comments
I think familiarity with proofs at the level of cs 20 or math 22 is enough. Even if you do not have the background, it is quite self studyable.
Proof writing background and basic probability knowledge.
experience in proofs and reading dense mathematic definitions
Definitely be confident in writing mathematical proofs. There isn't much CS background required, as almost all of the material in this course is new.
Some mathematical preparation; if you've never seen proofs it might seem daunting, but between pset0, the intro chapter, and the course staff, it's pretty easy to get caught up even if you started with nothing.
Some general CS knowledge

## Recommendations Comments - Would you recommend this course?

**What did you take away from your experience in this course? What did you learn? How did this course change you?**

Comments
One of the most difficult courses I've taken at Harvard. The course pushed me to think in new ways and I learned an immense amount.
This course is a great way to get exposure to some of the most fundamental areas in theoretical computer science. Boaz is a great lecturer, and the textbook, section, and office hours help a great deal in the learning process.
I had never considered theoretical computer science before taking this course, and although I learned a lot about it, I also learned that I do not find it particularly interesting. I have much more knowledge than I had going into this semester, but I don't think I'm going to use this new knowledge again.
I learned a lot and I was intellectually challenged and engaged.
Really cool to see how to classify all the problems that are out there. I learned to view computation in the abstract and really separate functions from implementations. Uncomputeable functions and complexity classes are fascinating.
The course is less practically useful, but it has important theoretical ideas that any computer scientist should know.
Theoretical CS is fun! More practice with reductions and understanding computability, complexity classes, and other key topics in TCS.
I feel this course gave me a very comprehensive foundation in theoretical computer science.
I learned how to rigorously reason through computational statements with math. I also learned how to engage with high-level complexity classes like NP, the hesitancies computer scientists have with statements about these classes, and the elegance of these ideas.
There is a lot to know about the inner workings of the tools we use in programming.
I gained a much better understanding of how computers work and the dreams of what they might one day do and the realities of what they can currently do.
Honestly, I enjoyed this course a lot more than I thought I would coming in. I always thought I was more interested in systems but theoretical CS is actually pretty fun. I definitely got more practice writing proofs, and some of the concepts I learned last year in CS 124 actually make sense (e.g. why you need to show both soundness and completeness). The Turing Machine stuff was interesting, and it was cool to see the overlap among the complexity classes.
I gained a better appreciation for theoretical computer science than I thought I would have.
I definitely was not interested in theoretical CS/TOC before this class, but I've started to find TOC really interesting and definitely plan to spend more time learning about and to take more classes related to TOC and theoretical CS more generally.
I really enjoyed the course, it was challenging but I now have a much more nuanced understanding of theoretical computer science.
This class teaches you all about models of computation, reductions, and time complexity classes. You'll be enlightened on how computation arose from circuits to DFAs to Turing machines and other cool Turing-complete models. You'll get plenty of experience with proofs and reading mathematical notation in the context of computer science. The experience you gain with reductions will definitely help with CS 124 (or vice versa if you already took CS 124). You'll also be very comfortable with time complexity classes and the open ended questions in theoretical CS. The course gave me more confidence in reading mathematical constructs, and is overall a wonderful course to take.
I learned a lot of important ideas behind computation and how to write strong proofs.
I left the course with a much better understanding of the theory behind computing and the fundamentals that allow us to use

Comments
computers to solve problems. I think it overall shaped my view on what it means to study theory in relation to computer science.
This course taught me that there exist professors who care more about their own image and career than their students. I learned almost nothing from the material and am further discouraged from taking any more computer science classes at Harvard.
I have learned *a lot* because of this course. It was a very nice introduction to TCS and it helped me a field I have previously never considered pursuing a career in.
Definitely really made me a lot better at proofs
I learned a lot in this course, I have a much greater understanding of introductory theoretical science and think that I can apply this into the future in other courses and even in practical computer science. Also intellectually I really enjoyed!
Theoretical CS is a really interesting field that I feel I did not appreciate as much as I do now, prior to this course and overall the theoretical CS department at Harvard.
Looking back, I think I learned a lot. I am not sure how useful this will be, but I can now confidently reason about different complexity classes etc.
Learned a lot about algorithms, and made me more comfortable with proof writing and abstract CS ideas.
I learned that computer science may not be the concentration that I want.
This was one of the coolest courses that I've ever taken. The open-questions that you learn about in this course are fascinating, and they make students excited about theory!
This course taught me how to grasp the big picture in theory in a systematic way. Turing Machines, reductions, and quantum computing—they're all rooted in the fundamentals of CS, such as NAND circuits, uniformity, and data representation. The course really teaches you how to appreciate what we do and don't know about CS.
Turing machines are the right way to think about the complexity of a task or (combined with information theory) the expressibility of a message. Godels Incompleteness theorem is INSANE. I now know sort of what quantum computing and cryptography are. I know how to do basic things with algorithms, like reductions. I know what P vs NP is.
The course really pushed me to learn more about how to write proofs and other concepts in theoretical computer science.
N/A
I think it improved my proof-writing skills and my problem solving skills in the computer science domain.
I learned a good amount about the limits of computation, and what we can and can't expect computers to do.
It was really nice to finally have complexity classes rigorously defined.
Everything is deeper than it seems. Knowing will help you do more effectively

### What would you like to tell future students about this class? (Your response to this question may be published anonymously.)

Comments
I enjoyed this course. The material is interesting; you learn about what it means to compute and when something is and is not computable. If you are interested in theoretical CS or are curious about how mathematically-rooted concepts can apply to computation in the real world, I encourage you to take this course.
This class is hard!!! Note that because 121 is no longer a CS concentration requirement, it is mostly a self-selected group of students choosing to enroll. Psets take a significant amount of time/office hours (especially if you are less familiar and have less background), and the two midterms and final are also all difficult and require a lot of studying.
This class is hard and is the only course up to this point that I have truly disliked. I wish I had taken CS 120 instead. The lectures are often messy as Boaz's teaching can sometimes jump all over the place, but make sure you read the textbook closely. Also, your contributions on Ed and writing public annotations in the textbook are part of your grade, so be prepared for that.
This class is pretty fun, but if you are a very strong math student already, you may get a bit bored from time to time. The course seems more geared towards CS concentrators who might want to take a look at the theoretical side of things than towards math people. If you're a CS person and you find the topics on Homework 0 (public on the course website) interesting, I think this class is a great introduction to the world of CS theory.
I really liked this class! I'd heard mixed reports (and seen the previous years' Q guides), but I had a great time. The material is super interesting, the psets/midterms/final were totally fair and manageable, and there's a ton of extra credit so there's very little grade pressure. Prof. Barak is a great instructor, and he clearly cares a lot about the students. There's a lot of reading, but the book is written specifically for the course, and you quickly develop a sense of what can safely be skipped. My one issue with the class is that expectations for participation were not clearly conveyed. The syllabus made it seem like section was optional, but they took attendance at section/OH, and similarly lecture was not mandatory but if you didn't go you were expected to participate in other ways. Also, we were graded on our Perusal participation, but nobody ever wrote anything so it was hard to participate (I generally think that Perusal is a bad idea, but I digress).

Comments
<p>I do think that this is not a class that most sophomores should be taking, and I feel like I got a lot more out of it by waiting until senior fall. It requires a fair amount of mathematical sophistication, and many of the bad Q reviews from previous years are just people who were not prepared for a theoretical course like this. If you take this class as an upperclassman with CS 20, CS 124 and/or a bunch of math classes under your belt, and you're comfortable with proofs, then you'll be able to appreciate the beautiful subject that is TCS without the learning curve of proofs.</p>
<p>Finally, since this really is the only place to say this: it is a huge failure of Harvard's CS department that CS 221 is only offered every few years, and that CS 125 hasn't been offered in awhile. Don't get me wrong, I greatly enjoyed 121, but I really wanted to challenge myself this fall, and there was no more advanced way for me to fulfill this concentration requirement. A school of Harvard's caliber should be offering grad-level TCS classes (or at least one class) yearly so that students who want to push themselves in a more intense and advanced class are able to do so.</p>
<p>I took 121 and 124 for the theoretical CS requirements and would do it over again if I had to. 121 covers beautiful material, the problems sets and exams are not too difficult, and there is plenty of bonus points to take care of your grade (at least there was when Boaz was teaching it to us). An understanding of theoretical CS will help everyone, although perhaps not in the most obvious ways. The way of thinking and viewing multiple different processes in the world as computation is intriguing, and I hope, useful.</p>
<p>A solid course. Boaz is a good teacher and knows his textbook in and out. The homeworks improve understanding tremendously, and the exams are fair. Sometimes the material feels a bit mundane, but this is a great primer for later CS classes (e.g. 124) or a smooth follow-through if you've already taken it.</p>
<p>This class covers a lot of interesting content, but isn't very well structured. If you enjoy theoretical CS and formal reasoning, you'll probably enjoy this class. If you enjoy more practical and applicable sides of CS more, however, this class probably won't be very rewarding. If choosing between this class and 124, I'd recommend most people take 124; although 124 is harder, the concepts are more useful and applicable, and I think the class is better structured.</p>
<p>I'd advise to take CS 120 instead. This class may be harder, but the stuff you learn in this class will not be useful to you unless you definitely plan to go into research. It's cool to learn some of the fundamentals, but prepare to get a bad grade unless you are constantly at office hours</p>
<p>Disclaimer: If you don't like a lot of proofy math, if you don't like reading proof as opposed to them being explained, or if you don't like theory, you might find my opinions misleading and infuriating once you enroll in the class. You have been warned.</p>
<p>That being said, this class is an absolute gem. It covers a lot of fantastic content in computer science. It gives you a high-level view of all of the basics of computer science. I love it; probably my favorite class yet. You should take it if you like challenging yourself and learning new stuff. You must, however, expect to grind, don't let pset 0 and 1 mislead you into thinking the class will be easy.</p>
<p>Here are some of the cool questions discussed in this class.  Do you want to explain to your friends how all computers are the same, how DNA, in theory, is the same as your laptop?  Are you interested in knowing how anyone can compute all finite functions? Do you want to learn the cracks in math, that there are uncomputable functions despite having the best machines, and that there are unprovable true statements despite being true? At that point, are they even true?  Can you create unbreakable encryptions?</p>
<p>If you want to discuss those questions and more in-depth, this class is for you.</p>
<p>Good content that is covered in the course, first few psets are hard and then they get easier. Also, now that CS120 is a thing, CS121 self-selects for people who are very invested in learning theoretical computer science, so the mean on the exams was very high.</p>
<p>This course is very well-taught and very useful, but it is indeed very difficult. You need to be willing to invest a lot of time in this class, and the staff will invest in you just as much.</p>
<p>This is quite a difficult class, but it is very rewarding and you will come out of it with a fairly good amount of knowledge in theoretical CS.</p>
<p>It's a fair class! You get out what you put into it. And the exams and psets can definitely be a bit tough, so do your best to go to office hours and learn from your peers.</p>
<p>This is a rigorous but fulfilling class about theoretical computer science! It definitely is a good amount of work, but if you are interested in theory, it is pretty interesting. Course staff and the textbook are really good!</p>
<p>CS121 is a great intermediate intro class like Stat110. The book is well-written, Boaz is an engaging and thoughtful lecturer, and the material is interesting. If you have the pre-requisites, can write proofs, and have the time in your schedule for an additional pset class, CS121 is well worth taking.</p>
<p>I took CS 124, then CS 121. There definitely is some overlap between the two classes, but definitely not enough to skip one after taking the other.  Honestly, if I could go back I would have taken CS 121. I feel like it would have served as a more gentle introduction to some of the</p>

Comments
<p>techniques used when writing proofs of correctness for algorithms (e.g. showing completeness and soundness). CS 124 hit the ground running, and just assumed you either understood these already or that you would learn it on your own.</p>
<p>CS 121 is a very manageable undergraduate CS course. I highly recommend actually doing the readings before the corresponding lecture; I found that gave me the most complete understanding of each topic. The psets take some work but aren't that bad. There are also a lot of opportunities for extra credit, which makes things less stressful. Definitely give the bonus questions a try. Professor Barak is a decent lecturer, although I've had friends who didn't like his style.</p>
<p>This class is definitely easier than 124, and in fact becomes even easier after having taken 124 due to the overlap in material (NP reductions, though there's talk there might be less of that in 124 next year.) It would also be good prep for 124 for the same reason, and also proof-writing. Still, it's not an insignificant time commitment. The teaching staff was very helpful. Some of the problem sets and most of the lectures are very dry: Boaz is not a great or clear lecturer, but READ THE TEXTBOOK, seriously, it will tell you everything you need to know.</p>
<p>Like for most concentration requirements, I think there are a lot of divided opinions regarding CS 121, but it was honestly one of my favorite classes that I took this semester. I wasn't expecting to enjoy theoretical CS at all, but I found a lot of results/concepts from this class really intriguing and unexpected (e.g., Rice's theorem, Y combinators, PRGs, Turing machines, etc), especially as someone who came into this class without any exposure to theoretical CS. I'd definitely recommend at least considering taking 121 – I know that CS 120 has become a much more popular option now that enrollment for that class isn't capped, and while 120 probably does a better job of preparing you for tech interviews than 121, I think that CS 121 introduces you to some really interesting concepts that aren't covered in 120 (e.g., PRGs, etc).</p>
<p>That being said, 121 isn't without its own flaws/differences – unlike MIT's TOC course, CS 121 uses Boaz's textbook, which still contains some typos and takes an unconventional approach to TCS. I thought that starting with NAND circuits and generally taking a more circuit-based approach was interesting and insightful, but I know some people who say they really would have preferred the standard approach to TOC covered in textbooks like Sipser. I also felt a bit lost at the very end of the course when we covered quantum computing – I think it was covered a little too quickly and assumed a little too much from the students wrt how much they would know about quantum notation/theory.</p>
<p>I still think the pros way outweighs the cons though: 121 is really interesting to begin with, and it's clear that Boaz cares a ton about his students and how well they learn the content in the class. There's so much extra credit in the class on psets and exams that it's super easy to get a 100 under psets (capped at 100) and its very doable to get 100+ on exams. CS 121.5 sections (where they invite profs and postdocs and the like to give related lectures) are super interesting (would definitely recommend trying to go to a few, even if you're unsure if you'll understand it all!), and 121 sections in general are really useful for pinning down the content. Problem sets are also interesting – they do a really nice job of tying course content together, and the results can be really insightful.</p>
<p>If you're on the border of trying 121 out, I'd go for it! Definitely would have regretted not taking it this semester!</p>
<p>If at all possible, avoid taking alongside 61. Will require a lot of work if you want to do really well but tbh not so much if you are aiming for B/B+ range. Other than midterm 1, the exams were pretty fair. Psets were lowkey impossible unless the tf tells you the answer at office hours – not sure what the rationale behind that was but it makes office hours pretty important. Content is not that interesting unless you are in to that kind of stuff, but tbh I would just pick 120 over this. I assume 120 better prepares you for 124.</p>
<p>CS 121 is an excellent course, and Boaz is an excellent instructor. His lectures are always clear, and above all, he is a kind and engaging teacher. The concepts and ideas you'll learn in this class are quite elegant (in my opinion), and I think the class does a good job of being both accessible and also offering students the chance to pursue topics in further depth.</p>
<p>If you took 124 before this, this will be a walk in the park. The first half of the class is a lot harder than the second half.</p>
<p>I took this class after CS124, but I would recommend taking it before as the course introduces some techniques (reductions, proofs of correctness and soundness) which can be useful preparation for 124 and other upper-level CS classes. I don't know how it compares to CS120, but the class is very well-designed and the material is interesting, so I would definitely recommend it to any CS concentrator.</p>
<p>This class teaches you all about models of computation, reductions, and time complexity classes. You'll be enlightened on how computation arose from circuits to DFAs to Turing machines and other cool Turing-complete models. You'll get plenty of experience with proofs and reading mathematical notation in the context of computer science. The experience you gain with reductions will definitely help with CS 124 (or vice versa if you already took CS 124). You'll also be very comfortable with time complexity classes and the open ended questions in theoretical CS. The course gave me more confidence in reading mathematical constructs, and is overall a wonderful course to take.</p>
<p>Please make sure that when you take this class that you make it your main focus of the semester. It is one of those courses where you can't really treat it like a side dish; it's a main entree course and it'll require a lot of your attention/focus.</p>
<p>As a concentration requirement, this class can be what you make of it. The assignments require work, and the material is not necessarily easy to grasp, but overall the class is pretty reasonable. If you are more inclined towards the theory side of things, then the material is honestly interesting, though you do not program at all. It's certainly a very different side of computer science, but one</p>

Comments
that is legitimately quite important.
This class has an unreasonable participation requirement, a useless textbook, incompetent TAs, and a professor who does not care enough to supervise his own exams.
Do the readings before psets and start psets early! The course isn't super hard, but it can get time consuming. The exams are non-trivial but easily doable given one actually studies for them. The teaching staff is amazing and the course structure is decent. Take this class if you want to meet cool people!
Read the textbook and go to office hours and section (lectures suck)
Interesting class with good support! Although challenging, if you engage with all the material you can learn a lot!
This course is difficult. It was not as hard or time consuming as CS 124, but still one of the hardest courses I have taken. Boaz made sure there were opportunities to make up for poor performance on some assignments or exams— there were ample opportunities for bonus points (do the reflections!) so it was not hard to get a 100 pset score when my average score on the psets (ignoring bonus) was in the low 90s.
You'll likely take this course for concentration requirements (particularly if you don't want to take CS 120). Depending on how interesting you find theoretical computer science as a field, you'll be more or less interested in the material this class covers. Regardless, expect this class to be a good deal of work — weekly psets! — and the material to be not so intuitive. At some point in the course, you'll probably end up going "maybe everything is unknown." It's unclear whether I'll retain any knowledge from this course in a year.
Boaz isn't the best lecturer but he is really passionate about the material and I think the class improved a ton since Mitzenmacher (at least from what I have heard). Be aware that there seems to be a selection bias amongst students who chose to take this class over CS120, so the exams medians tended to be extremely high (for Midterm 2 it was above 100% with EC). Psets can be very cumbersome to Latex but you will walk away with a rigorous understanding about what can be computed and complexity classes.
CS 121 is an interesting class with a lot of material. The concepts in the book are very deep and thought-provoking. If you put a lot of effort and time into the class, you will learn a lot, especially with optional advanced sections CS 121.5 to enhance your understanding of theoretical computation. That being said, the problem sets and exams are not demanding, suggested by the over 100% average in the midterm.
Please go to office hours and meet with the Patel fellow. They will help you immensely. This class is definitely a grind and will have you question your very existence (or at least, your concentration) but believe me when I say, you'll eventually make it. Just dedicate your time to consistently watching lecture, completing the readings, starting PSETs EARLY, finding a PSET group (!!!!!), and going to office hours. Boaz is one of the kindest CS professors you'll find at this institution; befriend him and go to lectures to ask clarifying questions; he'll appreciate you for that :) Regardless, if you have a group of people that you can work with, you'll suffer together, yes, but you'll persevere and get your degree together too! If you're ever in a dark place while taking this class, just take a deep breath, do something to clear your mind, and remember that this class (nor any of your classes) define your worth. You've got this <3
121 is hard, but not impossible. Boaz really does care about his students and is a great resource, and there's so much material he's made that you can find what works for you (I'd say don't kill yourself over the textbook if you can understand the material through lectures or practice problems instead). The psets can get kind of tedious, but there's also a lot of room for extra credit that takes off a lot of the pressure. You probably have to do it so might as well make the most of it
Please do not take this course! I had to take it for my concentration requirement. Any other course is more helpful than this. If you're interested in the topic, please take CS 120 or 124, BUT NOT THIS COURSE! The professor, Boaz, did show excitement toward the topic, but lectures went very slowly and oftentimes were spent going over proofs rather than seeing applicability. Sections were not helpful since the TFs would only go over the textbook content and were no practice problems. OH was oftentimes a waste of time since either TFs assumed you knew the material perfectly from the textbook or were kind of mean when you had more questions. The only ones who made this class a little easier were Gabe and Elijah, who actually taught the material well and made me get some use out of the class. Also, the textbook and psets had a ton of mistakes and made it sometimes difficult to complete PSETS on time. So, please don't take this course if you do not have to. You'll save yourself a semester of depression.
If you've ever heard about the P=NP debate and its implications for daily life and became more interested, then this is the course for you!
Take CS121! The course is very well-structured (lots of support from the teaching team) and Boaz clearly puts a lot of effort into the course
I absolutely recommend taking this course. Even if you might not have previous theory experience or interest, the topics of this course are fascinating and truly super exciting. The opportunities to improve your understanding via bonus points on the problem-sets / reflections are wonderful, encouraging true understanding! If you are worried about the difficulty, make sure to read the textbook, as it is super friendly and clear! I highly recommend taking notes and reading every chapter a couple of times! One of the coolest classes I've ever taken, and truly one of the most valuable courses at Harvard!
Always read the textbook and go to section. The textbook is a nitty-gritty dive into the details behind the concepts; classes and sections provide intuition for thinking about high-level concepts. The course clicked for me once I started focusing more on big-picture intuition—e.g. the relationship between finite and infinite—and less on the details, e.g. how we can make a Turing Machine oblivious (take the course to find out what that means!).

Comments
Super fun content sometimes (not always). Lots of extra credit. Psets are no slouch but also not too bad. Good support from the staff. Goes fast sometimes.
I was choosing between this class and AM 121 (Optimization) for fulfilling an Applied Math Discrete Math requirement. This class was certainly more challenging. A good group of friends taking the class makes a big difference. Problem sets and exams are quite challenging, but there are ample opportunities for bonus points so don't feel too stressed about that. Boaz wasn't my favorite lecturer, but the textbook is comprehensive and helps a lot. If you're in my position where you don't *have* to take this class but are interested in a challenge, I would recommend giving it a try. If I were to go back, I would probably make the same decision, but for sure there were times throughout the semester where the challenge felt tough.
Boaz is a wonderful professor who cares deeply about the class and the course material, and this comes through in every lecture and textbook chapter. The textbook is extremely thorough, and the PSETs are very manageable going to one or two office hours per week.
I would block out a large amount of time in your schedule and try to balance CS121 with easier courses.
Only take it if you have to.
Start problem sets early and do the reading. Make sure to review past problems for exams
Very easy if you've taken Cs124 previously. Quite grade-inflated this year, there is a lot of extra credit available on the psets and even the exams.
I think this class was a solid introduction to theoretical CS (as the name suggests) and the difficulty level was quite manageable. It wasn't extremely time consuming, but the concepts we learned were very interesting and important. It covers the classical TCS topics like circuits, automata, turing machines, uncomputability, reductions, and P/NP/EXP, and probabilistic computation. I thought this coverage was very comprehensive, and I quite enjoyed it. At the end, we did some special topics including pseudorandomness, crypto, and quantum. These were also quite interesting. I think this course is a great balance between interesting content/manageability.
I was a simultaneously enrolled student, I definitely recommend it if you have another course you want to take. There's so much material that you can self-study the whole course. You can watch lectures at increased speed and study the readings, and you will be fine on the psets. Once you do the psets, the exams are very manageable. I took this class just because it's a requirement for the computational limitations tag, but i think it was worth it. I saved a lot of time because of simultaneous enrollment as I didn't need to trek to the SEC and watch the lectures whenever I wanted. I strongly recommend this!
Sometimes though, the comments of graders felt a bit insulting. They sometimes act like we have no experience with any of the materials and assume little background, and they can deduct points for random things that are obvious. This was especially a problem with midterm 2 where they gave 0 points for solving a problem in a different way than they wanted, and wouldn't accept any regrade requests. The answer they gave was very dismissive and reflected a grading philosophy based on punishment, rather than giving meaningful feedback. I think this can be a problem in general in CS where people assume that others do not know as well and interpret different ways of reasoning as flawed. As a CA for a math class, I always try to give very comprehensive feedback when I am going to deduct a large amount of points, and I try to make sure I am not misunderstanding the student. I think this is the philosophy they should have embraced. Other than this, I think the course was great, and this fortunately does not affect your experience too badly. There are so many bonus points that the second midterm average was something like 103 (the class is not curved). So, you'll be fine.
Do the readings (you might skip the main body of the proofs but at least read the "proof idea" parts), attend lectures/ watch them. Start the problem sets early. Try to do the problem sets on your own before seeking help in office hours to understand the material deeply and improve your proof writing skills.
If you are really interested in learning more about theories of computation and what computers can and can't do, then consider taking this class. Otherwise, I think taking CS 120 may be a better use of time. This course itself is fairly paced, and you spend enough time on each unit to really understand it well.
121 is a good intro to theoretical CS, surveying circuits, Turing machines, finite automata, complexity classes, randomized algorithms, and some time on cryptography and quantum computing. It's not overly challenging, but still keeps you engaged; the assignments are all proof-based.

## Instructor Comments

**Please comment on this person's teaching. (Your response to this question may be published anonymously.)**

Comments
Boaz's lectures seem like they could be slightly better prepared, as there are fairly often mistakes in his slides that he needs to correct as he's lecturing. Also, his lecturing kinda jumps around all over the place, so it's sometimes hard to follow.



Comments
I wasn't a huge fan of Boaz's lectures, since they moved a little too slowly for my taste, but his enthusiasm for the course and theoretical computer science in general was clear; I think he was very present and a useful resource to some people who were struggling a little more in the course.
Prof Barak is a legend — super excited about the material, a great lecturer, and cares a ton about students.
Sometimes lectures felt a bit chaotic, like there was too much packed in, but most of the times it was good.
Boaz is a passionate teacher. His understanding of the material is exceptionally deep, and the only times he struggles is when he gets overly ambitious and tries to cover too much material (looking at you, ML lectures) and instead only skims unsatisfactorily on the surface of content. Overall, however, I liked him a lot.
Lectures were pretty good but it was easy to get lost at times
One of the best lecturers that I've had in my CS courses at harvard
I was incredibly impressed by how invested Prof. Barak was in the class, even with so many TFs available for helping students. Prof. Barak would respond to questions on Ed regularly, and respond to questions on Persuall within HOURS. I have honestly never been in a CS course like this, and it really helps.
Boaz was a very good lecturer, and his lectures were pretty engaging and helpful in learning the course material.
Boaz is a caring guy who seems genuinely passionate and concerned about the CS topics at hand. His explanation of things can take a bit of time to wrap your head around but to be fair this is probably also due to the advanced material. Overall a very decent professor!
Boaz is an excellent lecturer, and I really enjoyed his class. It seems that he really cares about his students, and his lectures are always engaging, interesting, and light-hearted.
Professor Barak does not know how to teach a STEM class. He forces his students to read his own textbook and then attend lectures which provide no enhancement. He does not care about students understanding the material and is only obsessed with assigning point values to them.
Very enthusiastic, passionate about what he's teaching. The lectures weren't always easy to follow (professor Barak has a very distinct teaching style), but I enjoyed coming to them. He encourages participation and answers all the questions we ask.
I just really think he sometimes skips many steps in his explanations in lecture and sometimes the logic he gives verbally is confusing
Prof. Barak is a wonderful lecturer! He's humorous and is genuinely excited about his teaching, which shines through. Even in lecture, he creates a welcoming culture of asking questions and clarification, which I didn't get often in other CS classes.
Professor Barak is passionate about what he is teaching. The book he wrote is exceptional in terms of concept and also depth. Lectures could potentially be improved by bridging the gap between textbook and classroom.
When concepts are a bit confusing, he does a great job of clarifying points of confusion that people bring up in lecture or Perussal. He's also just very kind and honest, which is refreshing in the CS department. His enthusiasm for the material makes going to class more pleasant even though I'm not too keen on theoretical CS
He's very passionate about the topic, but sometimes his lectures are a bit windy and dry. It's definitely gotten a lot better from past years though! And he's super accessible so would highly recommend just going to chat with him in OHs.
I loved Boaz — he really loves teaching this course and you can tell!
Boaz is an incredible teacher and lecturer! Every lecture is exciting and fascinating, and his passion is truly contagious about the subjects! There are a ton of opportunities to better understand material, and Boaz is also super kind and willing to help! The class is very well organized and incredibly well taught.
Blended humor and clear explanations to effortlessly explain difficult concepts!
There's a good balance of philosophy and practice in his lectures. They are sometimes too fast to follow well (particularly towards the end of the course). Boaz is super active Ed and Perusall, and he has regular office hours.
Boaz wasn't the most effective lecturer, with his style of teaching not being very linear. As such it was hard to follow lectures at times. Overall he did a good job, though.
Great teaching
Incredible textbook writer and great lecture slides, but poor lecturer.
Professor Boaz gives effective lectures with slides, but sometimes when he writes on the board to explain something, I think he often confuses students because it's not that well organized. Other than that, he's a very great professor that builds a lot of enthusiasm about theoretical computer science!
Professor Boaz is very clearly knowledgeable and enthusiastic about this course. However, his lectures can be somewhat slow at times. It occasionally feels like he has not fully prepped the material, so he sometimes spends unneeded time thinking about how to explain the material. I definitely think additional work can spent streamlining these lectures.
Boaz is great, and clearly very excited about theoretical CS. Lectures could feel a little slow at times, but Boaz questions after class/

## Comments

in his office hours was always extremely worthwhile.

## Custom Questions

### Comments on the Patel fellow, and the Patel fellowship program in general

## Comments

He wasn't the most accessible because the offerings were in person in the quad. Not the most responsive

Very helpful.

This program was a lifesaver for me. Elijah was awesome.

I didn't personally interact with Elijah but it looks like he made an effort to engage with students based on his emails.

It is a wonderful and very helpful program. Elijah was very helpful for me this semester.

Elijah Tai provided the only useful help in this course.

Elijah is incredible: extremely kind, really knowledgeable and a skilled teacher – a wonderful combination. The Patel fellowship program has been an unparalleled program for courses that I have found to be most challenging. I think it has really strengthened my foundational knowledge of computer science.

Literally the best program to have in the CS department! Elijah was absolutely incredibly. He's so knowledgeable about the course concepts and does an amazing job of explaining material I'm confused on. Simply remarkable. I appreciate all the time he dedicated this semester to helping me. He's the main reason I ended up improving significantly :)

Amazing resource! I would not have understood a single thing from this course without Elijah's help.

Didn't really engage with the Patel fellow.

Great. Helped me a lot and was always available.

It's so apparent that Patel fellows are necessary—I think there should always be a Patel fellow for large classes past CS51. It's a type of help that normal office hours have not been able to provide me, and a type of help that I wish I could take advantage of in all my years at Harvard studying higher level CS.

N/A

A great program with a great fellow!