Lecture 38

CSE 331

Dec 2, 2019

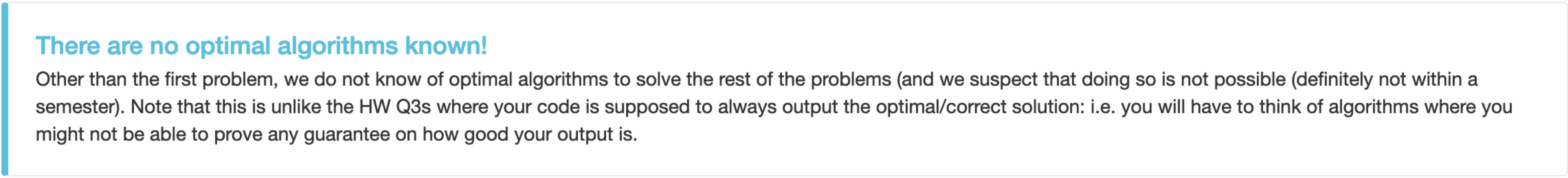
Quiz starts at 1pm and ends at 1:10pm

Q2: $c_e > 0$

Lecture starts at 1:15pm

Upcoming deadlines @11am

Problems 4,5: **FRIDAY**, Dec 6



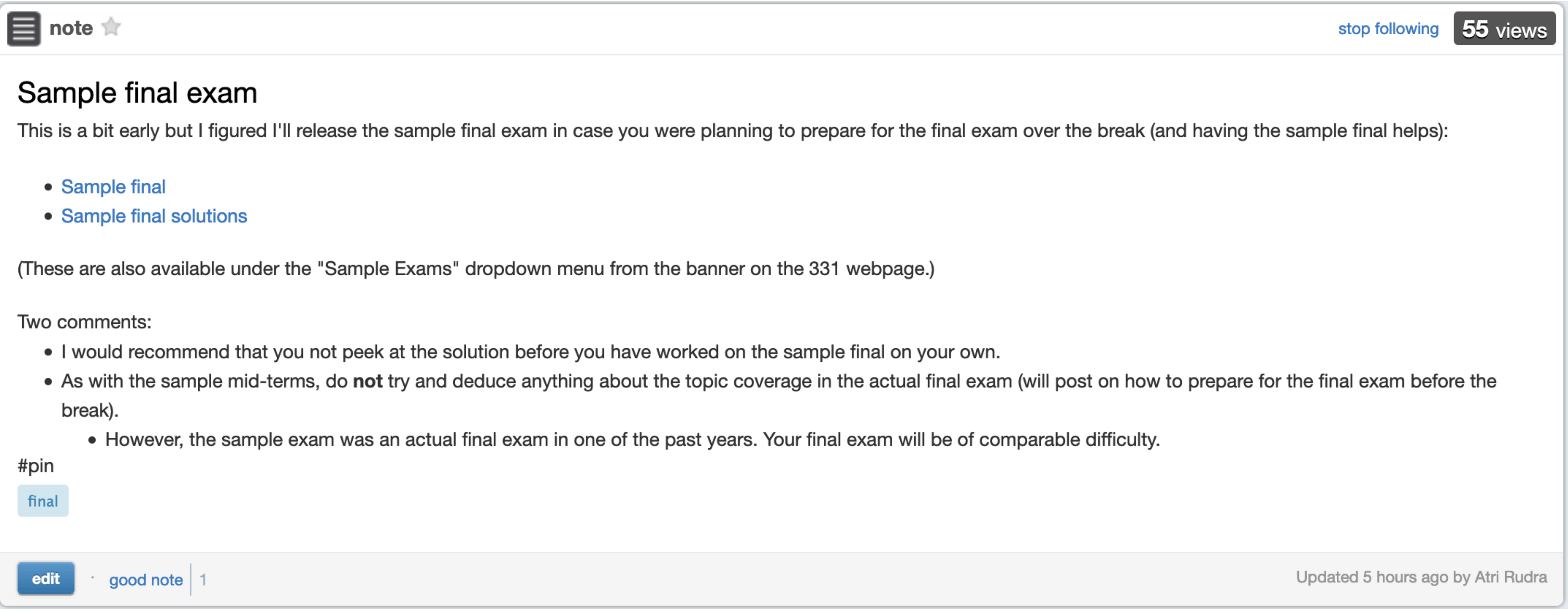
There are no optimal algorithms known!

Other than the first problem, we do not know of optimal algorithms to solve the rest of the problems (and we suspect that doing so is not possible (definitely not within a semester). Note that this is unlike the HW Q3s where your code is supposed to always output the optimal/correct solution: i.e. you will have to think of algorithms where you might not be able to prove any guarantee on how good your output is.

Try your solution on all Problems 4 to 5!

Make sure everyone has accept group invitation and THEN submit

Sample Final Exam



Sample final exam

This is a bit early but I figured I'll release the sample final exam in case you were planning to prepare for the final exam over the break (and having the sample final helps):

- Sample final
- Sample final solutions

(These are also available under the "Sample Exams" dropdown menu from the banner on the 331 webpage.)

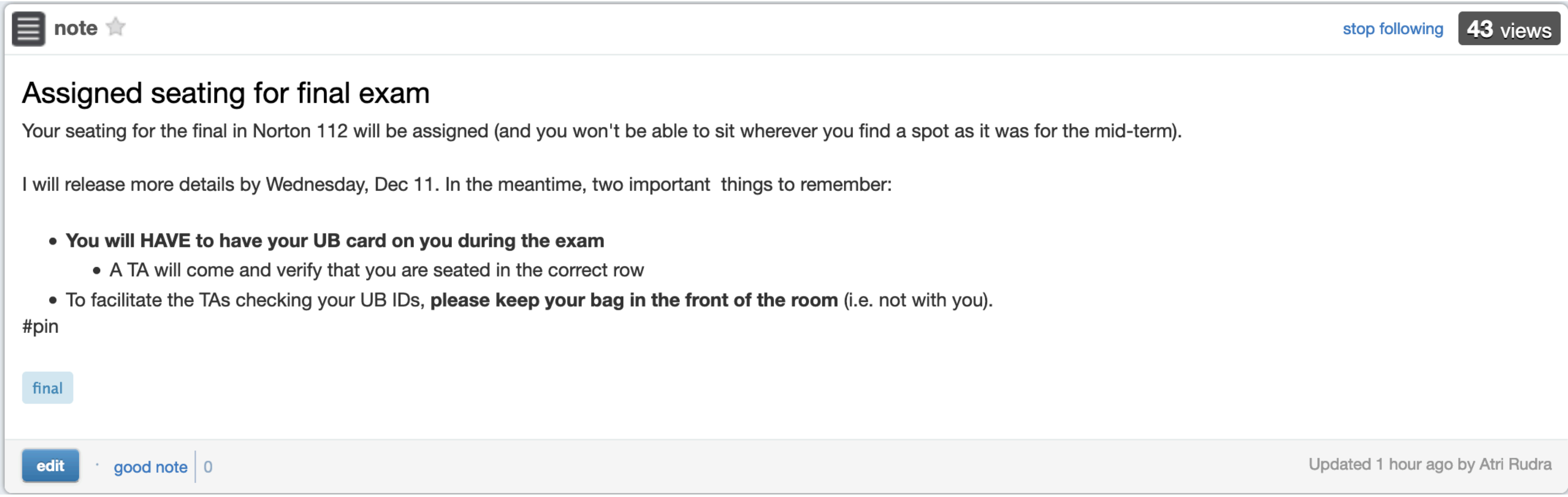
Two comments:

- I would recommend that you not peek at the solution before you have worked on the sample final on your own.
- As with the sample mid-terms, do **not** try and deduce anything about the topic coverage in the actual final exam (will post on how to prepare for the final exam before the break).
    - However, the sample exam was an actual final exam in one of the past years. Your final exam will be of comparable difficulty.

#pin

Updated 5 hours ago by Atri Rudra

Final exam seating



Assigned seating for final exam

Your seating for the final in Norton 112 will be assigned (and you won't be able to sit wherever you find a spot as it was for the mid-term).

I will release more details by Wednesday, Dec 11. In the meantime, two important things to remember:

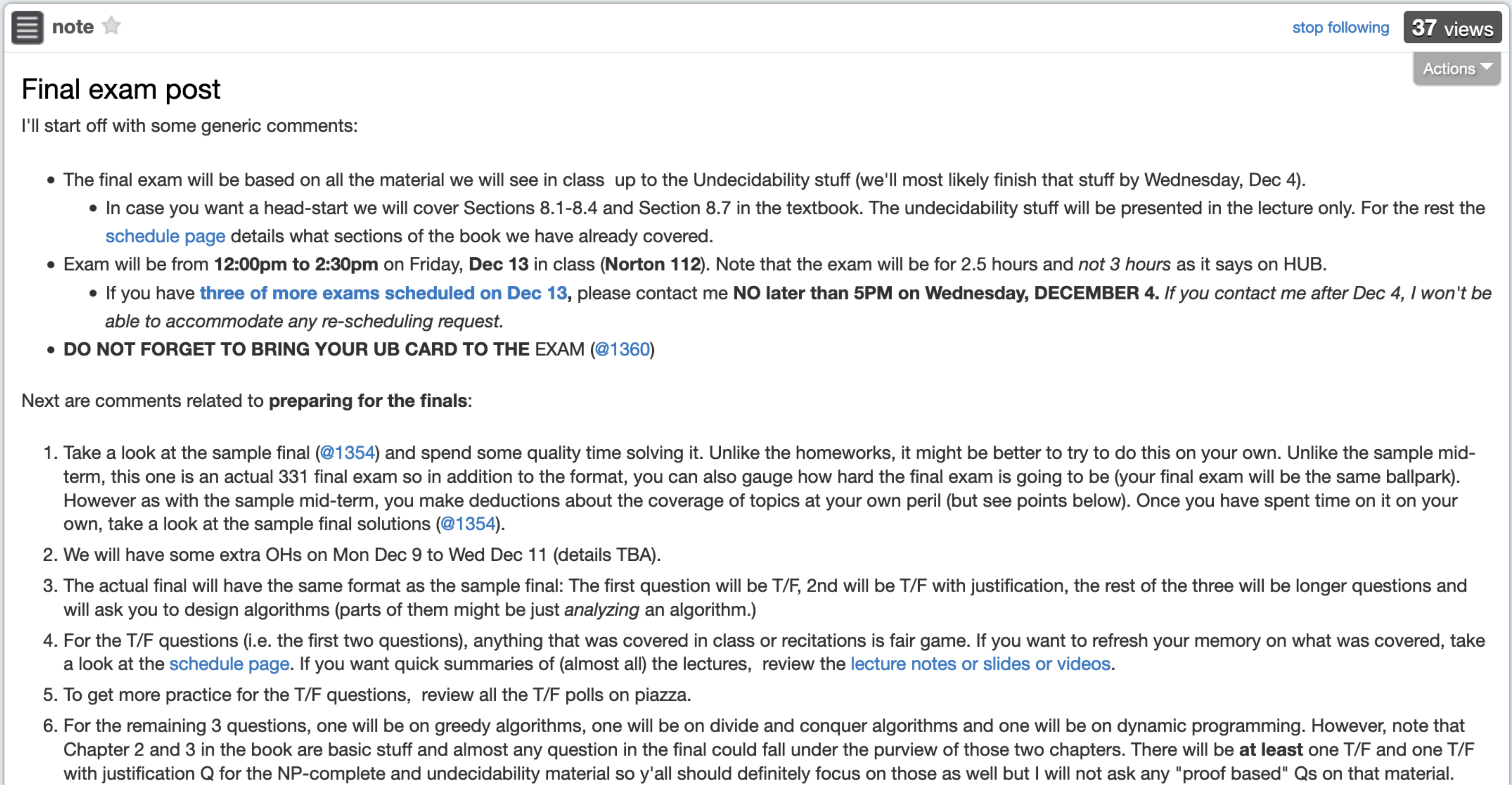
- **You will HAVE to have your UB card on you during the exam**
  - A TA will come and verify that you are seated in the correct row
- To facilitate the TAs checking your UB IDs, **please keep your bag in the front of the room** (i.e. not with you).

#pin

final

Updated 1 hour ago by Atri Rudra

The final exam post



Final exam post

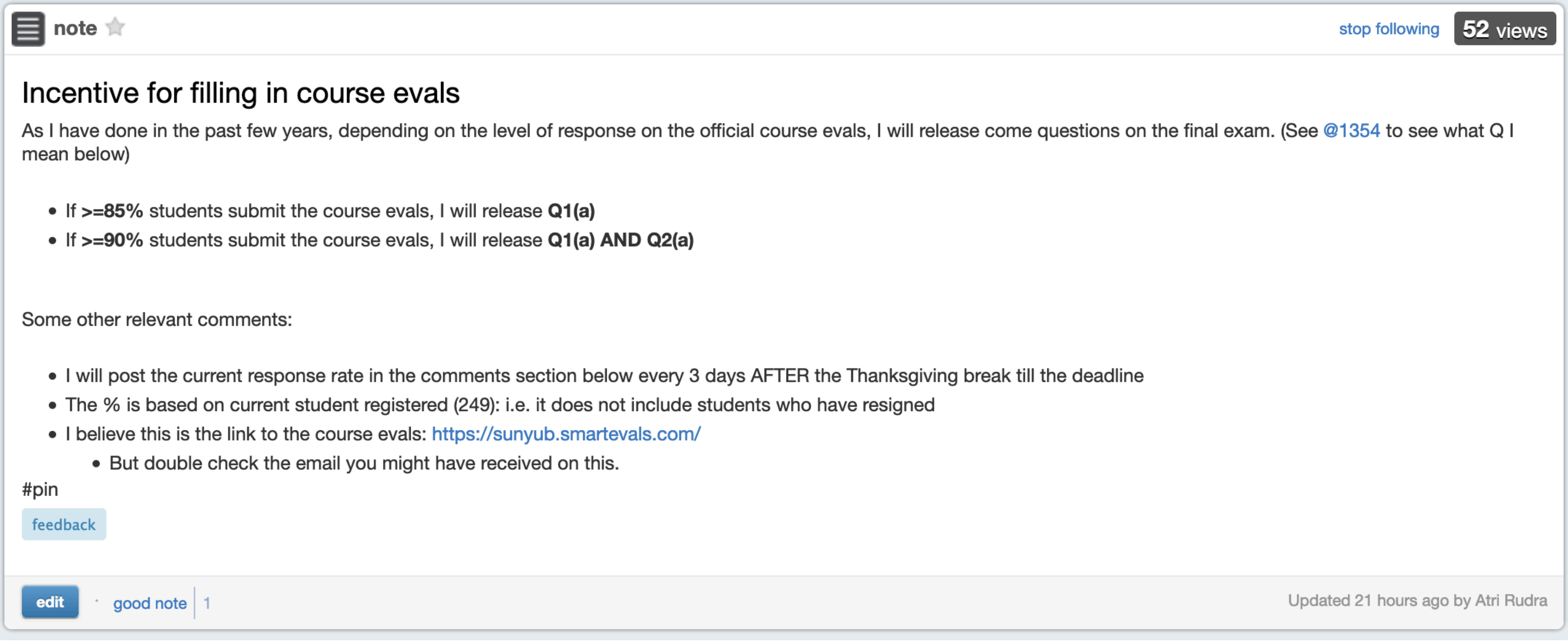
I'll start off with some generic comments:

- The final exam will be based on all the material we will see in class up to the Undecidability stuff (we'll most likely finish that stuff by Wednesday, Dec 4).
    - In case you want a head-start we will cover Sections 8.1-8.4 and Section 8.7 in the textbook. The undecidability stuff will be presented in the lecture only. For the rest the schedule page details what sections of the book we have already covered.
- Exam will be from **12:00pm to 2:30pm** on Friday, **Dec 13** in class (**Norton 112**). Note that the exam will be for 2.5 hours and *not 3 hours* as it says on HUB.
    - If you have **three of more exams scheduled on Dec 13**, please contact me **NO later than 5PM on Wednesday, DECEMBER 4.** *If you contact me after Dec 4, I won't be able to accommodate any re-scheduling request.*
- **DO NOT FORGET TO BRING YOUR UB CARD TO THE** EXAM (@1360)

Next are comments related to **preparing for the finals**:

1. Take a look at the sample final (@1354) and spend some quality time solving it. Unlike the homeworks, it might be better to try to do this on your own. Unlike the sample mid-term, this one is an actual 331 final exam so in addition to the format, you can also gauge how hard the final exam is going to be (your final exam will be the same ballpark). However as with the sample mid-term, you make deductions about the coverage of topics at your own peril (but see points below). Once you have spent time on it on your own, take a look at the sample final solutions (@1354).
2. We will have some extra OHs on Mon Dec 9 to Wed Dec 11 (details TBA).
3. The actual final will have the same format as the sample final: The first question will be T/F, 2nd will be T/F with justification, the rest of the three will be longer questions and will ask you to design algorithms (parts of them might be just *analyzing* an algorithm.)
4. For the T/F questions (i.e. the first two questions), anything that was covered in class or recitations is fair game. If you want to refresh your memory on what was covered, take a look at the schedule page. If you want quick summaries of (almost all) the lectures, review the lecture notes or slides or videos.
5. To get more practice for the T/F questions, review all the T/F polls on piazza.
6. For the remaining 3 questions, one will be on greedy algorithms, one will be on divide and conquer algorithms and one will be on dynamic programming. However, note that Chapter 2 and 3 in the book are basic stuff and almost any question in the final could fall under the purview of those two chapters. There will be **at least** one T/F and one T/F with justification Q for the NP-complete and undecidability material so y'all should definitely focus on those as well but I will not ask any "proof based" Qs on that material.

Course evaluations incentive



Incentive for filling in course evals

As I have done in the past few years, depending on the level of response on the official course evals, I will release come questions on the final exam. (See @1354 to see what Q I mean below)

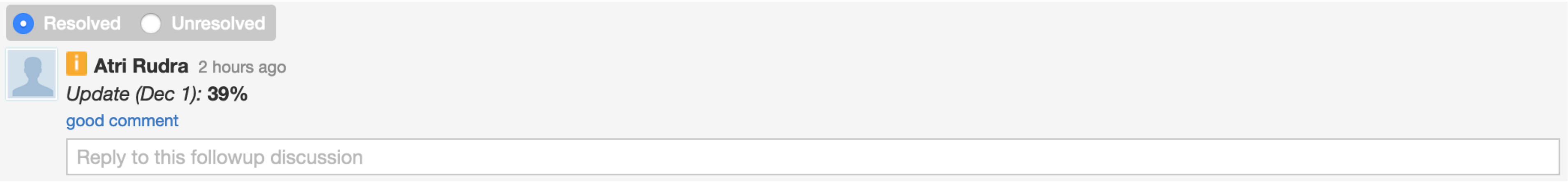
- If **>=85%** students submit the course evals, I will release **Q1(a)**
- If **>=90%** students submit the course evals, I will release **Q1(a) AND Q2(a)**

Some other relevant comments:

- I will post the current response rate in the comments section below every 3 days AFTER the Thanksgiving break till the deadline
- The % is based on current student registered (249): i.e. it does not include students who have resigned
- I believe this is the link to the course evals: https://sunyub.smartevals.com/
  - But double check the email you might have received on this.

#pin

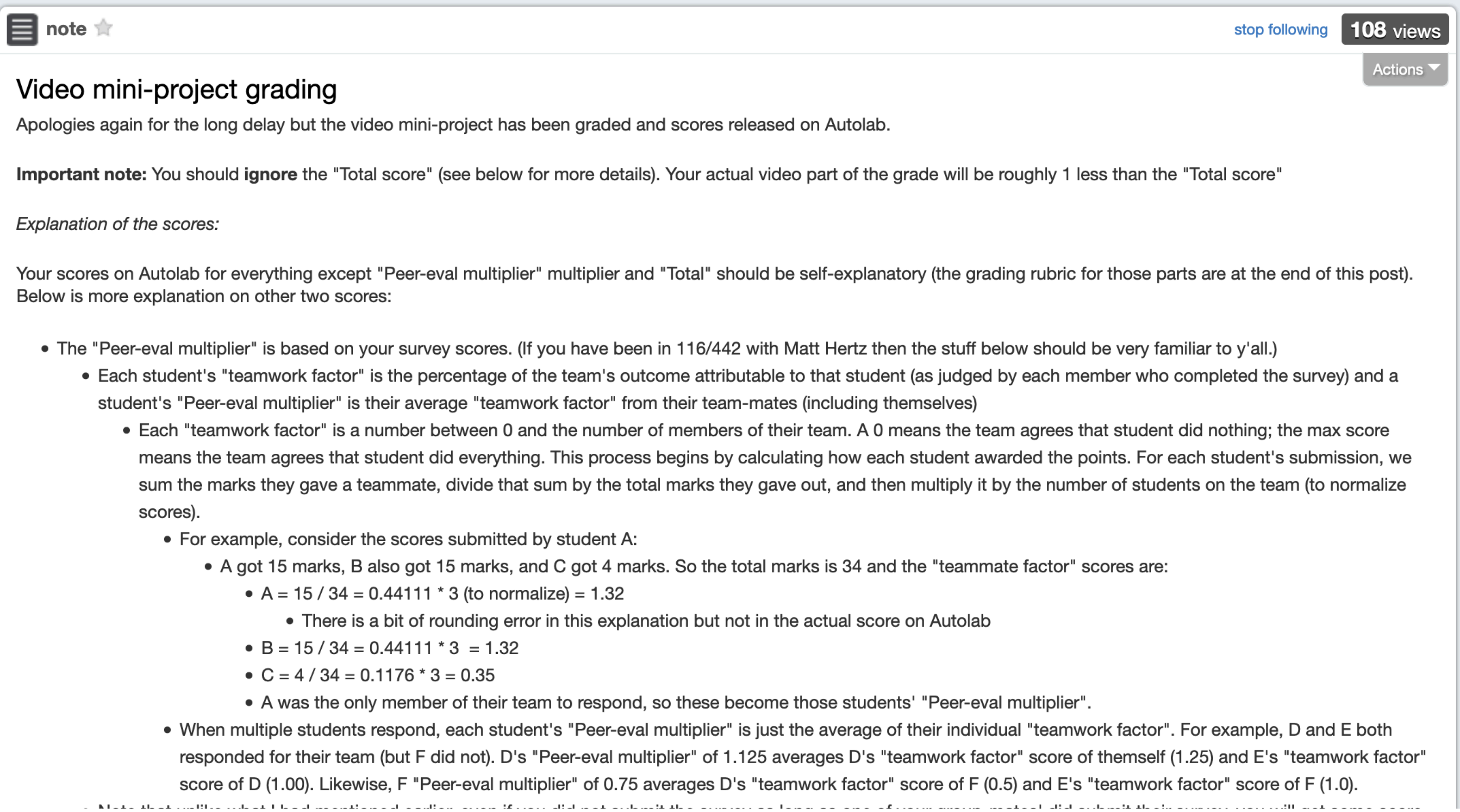
Updated 21 hours ago by Atri Rudra



Atri Rudra 2 hours ago

Update (Dec 1): **39%**

Video mini-project graded



Video mini-project grading

Apologies again for the long delay but the video mini-project has been graded and scores released on Autolab.

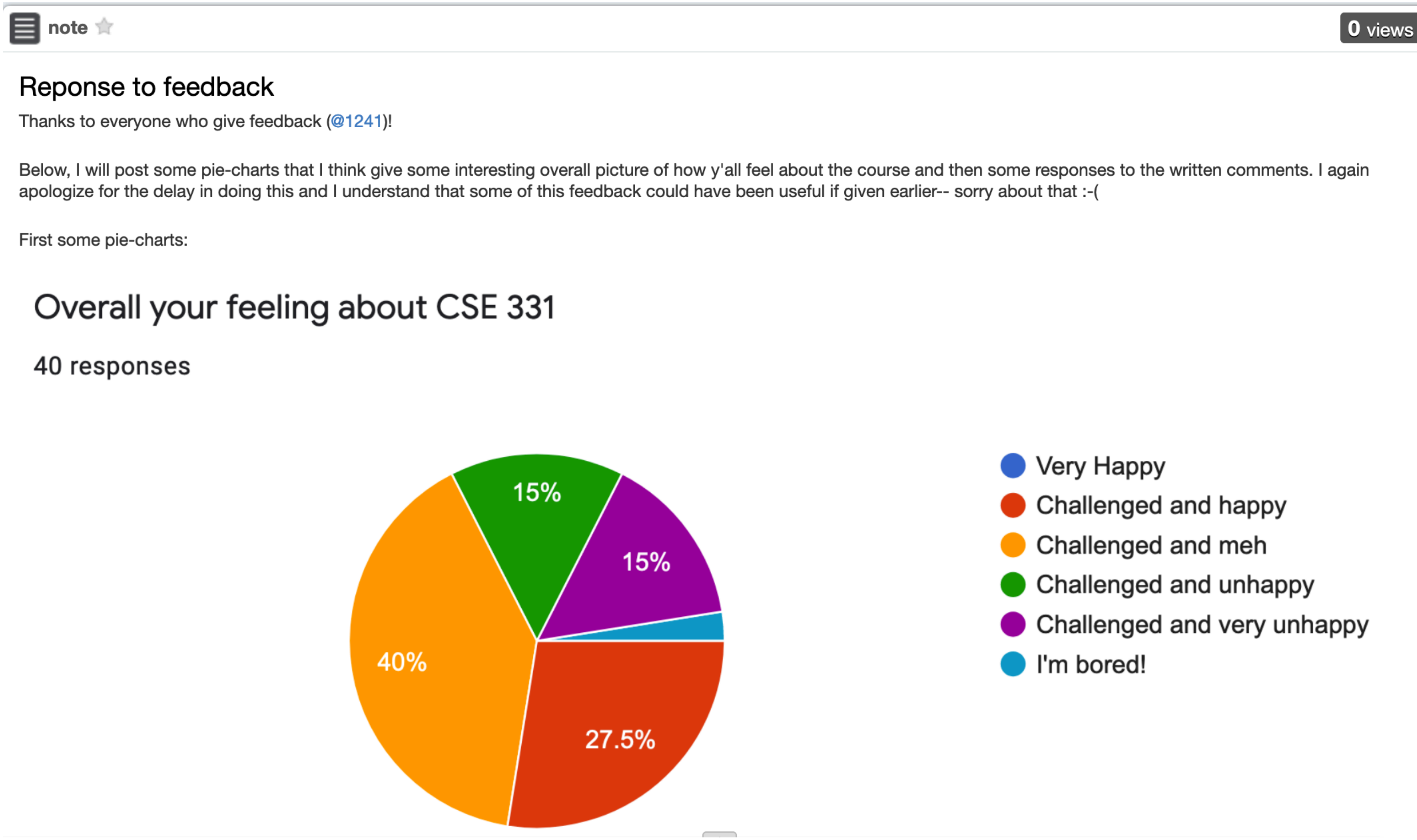
Important note: You should **ignore** the "Total score" (see below for more details). Your actual video part of the grade will be roughly 1 less than the "Total score"

Explanation of the scores:

Your scores on Autolab for everything except "Peer-eval multiplier" multiplier and "Total" should be self-explanatory (the grading rubric for those parts are at the end of this post). Below is more explanation on other two scores:

- The "Peer-eval multiplier" is based on your survey scores. (If you have been in 116/442 with Matt Hertz then the stuff below should be very familiar to y'all.)
    - Each student's "teamwork factor" is the percentage of the team's outcome attributable to that student (as judged by each member who completed the survey) and a student's "Peer-eval multiplier" is their average "teamwork factor" from their team-mates (including themselves)
        - Each "teamwork factor" is a number between 0 and the number of members of their team. A 0 means the team agrees that student did nothing; the max score means the team agrees that student did everything. This process begins by calculating how each student awarded the points. For each student's submission, we sum the marks they gave a teammate, divide that sum by the total marks they gave out, and then multiply it by the number of students on the team (to normalize scores).
            - For example, consider the scores submitted by student A:
                - A got 15 marks, B also got 15 marks, and C got 4 marks. So the total marks is 34 and the "teammate factor" scores are:
                    - A = 15 / 34 = 0.44111 * 3 (to normalize) = 1.32
                        - There is a bit of rounding error in this explanation but not in the actual score on Autolab
                    - B = 15 / 34 = 0.44111 * 3 = 1.32
                    - C = 4 / 34 = 0.1176 * 3 = 0.35
                    - A was the only member of their team to respond, so these become those students' "Peer-eval multiplier".
            - When multiple students respond, each student's "Peer-eval multiplier" is just the average of their individual "teamwork factor". For example, D and E both responded for their team (but F did not). D's "Peer-eval multiplier" of 1.125 averages D's "teamwork factor" score of themself (1.25) and E's "teamwork factor" score of D (1.00). Likewise, F "Peer-eval multiplier" of 0.75 averages D's "teamwork factor" score of F (0.5) and E's "teamwork factor" score of F (1.0).
    - Note that unlike what I had mentioned earlier, even if you did not submit the survey as long as one of your group-mates' did submit their survey, you will get some score

Response to feedback



note ☆

0 views

Reponse to feedback

Thanks to everyone who give feedback (@1241)!

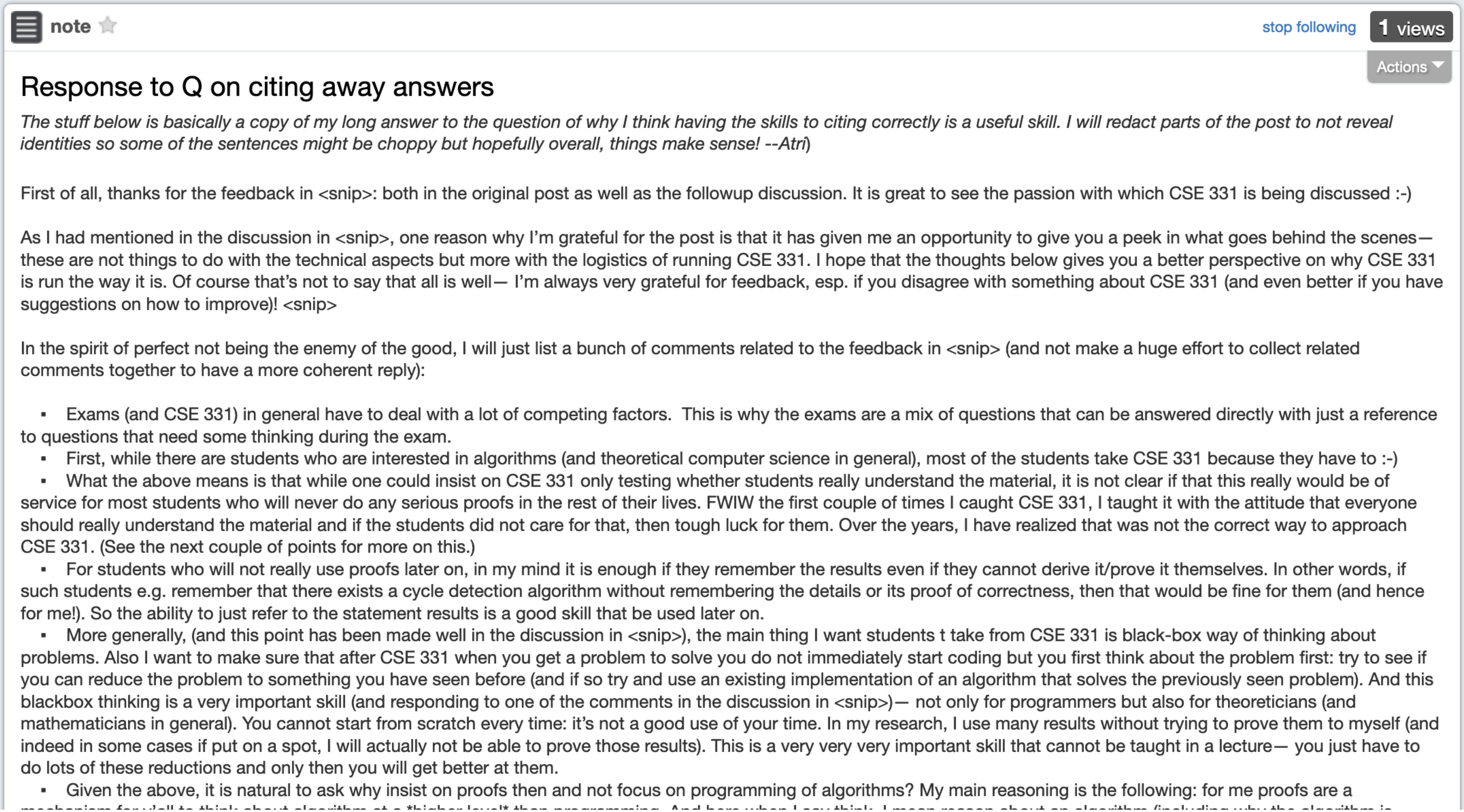
Below, I will post some pie-charts that I think give some interesting overall picture of how y'all feel about the course and then some responses to the written comments. I again apologize for the delay in doing this and I understand that some of this feedback could have been useful if given earlier-- sorry about that :-(

First some pie-charts:

Overall your feeling about CSE 331

40 responses

On citing away answers



Response to Q on citing away answers

The stuff below is basically a copy of my long answer to the question of why I think having the skills to citing correctly is a useful skill. I will redact parts of the post to not reveal identities so some of the sentences might be choppy but hopefully overall, things make sense! --Atri)

First of all, thanks for the feedback in <snip>: both in the original post as well as the followup discussion. It is great to see the passion with which CSE 331 is being discussed :-)

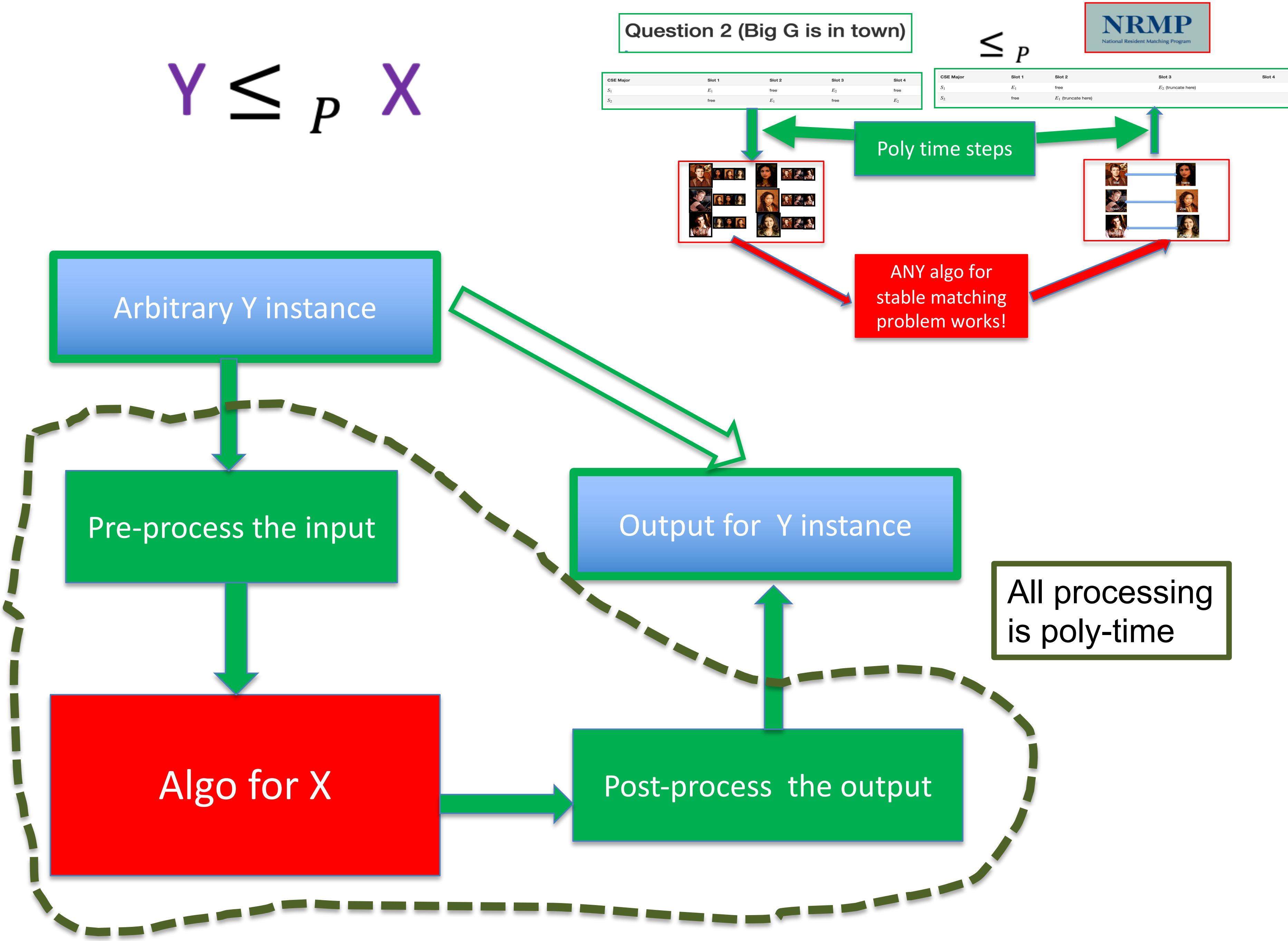
As I had mentioned in the discussion in <snip>, one reason why I'm grateful for the post is that it has given me an opportunity to give you a peek in what goes behind the scenes— these are not things to do with the technical aspects but more with the logistics of running CSE 331. I hope that the thoughts below gives you a better perspective on why CSE 331 is run the way it is. Of course that's not to say that all is well— I'm always very grateful for feedback, esp. if you disagree with something about CSE 331 (and even better if you have suggestions on how to improve)! <snip>

In the spirit of perfect not being the enemy of the good, I will just list a bunch of comments related to the feedback in <snip> (and not make a huge effort to collect related comments together to have a more coherent reply):

- Exams (and CSE 331) in general have to deal with a lot of competing factors. This is why the exams are a mix of questions that can be answered directly with just a reference to questions that need some thinking during the exam.
- First, while there are students who are interested in algorithms (and theoretical computer science in general), most of the students take CSE 331 because they have to :-)
- What the above means is that while one could insist on CSE 331 only testing whether students really understand the material, it is not clear if that this really would be of service for most students who will never do any serious proofs in the rest of their lives. FWIW the first couple of times I caught CSE 331, I taught it with the attitude that everyone should really understand the material and if the students did not care for that, then tough luck for them. Over the years, I have realized that was not the correct way to approach CSE 331. (See the next couple of points for more on this.)
- For students who will not really use proofs later on, in my mind it is enough if they remember the results even if they cannot derive it/prove it themselves. In other words, if such students e.g. remember that there exists a cycle detection algorithm without remembering the details or its proof of correctness, then that would be fine for them (and hence for me!). So the ability to just refer to the statement results is a good skill that be used later on.
- More generally, (and this point has been made well in the discussion in <snip>), the main thing I want students t take from CSE 331 is black-box way of thinking about problems. Also I want to make sure that after CSE 331 when you get a problem to solve you do not immediately start coding but you first think about the problem first: try to see if you can reduce the problem to something you have seen before (and if so try and use an existing implementation of an algorithm that solves the previously seen problem). And this blackbox thinking is a very important skill (and responding to one of the comments in the discussion in <snip>)— not only for programmers but also for theoreticians (and mathematicians in general). You cannot start from scratch every time: it's not a good use of your time. In my research, I use many results without trying to prove them to myself (and indeed in some cases if put on a spot, I will actually not be able to prove those results). This is a very very very important skill that cannot be taught in a lecture— you just have to do lots of these reductions and only then you will get better at them.
- Given the above, it is natural to ask why insist on proofs then and not focus on programming of algorithms? My main reasoning is the following: for me proofs are a mechanism for y'all to think about algorithm at a *higher level* than programming. And here when I say think, I mean reason about an algorithm (including why the algorithm is

Questions?

$Y \leq_P X$



Question 2 (Big G is in town)

CSE Major	Slot 1	Slot 2	Slot 3	Slot 4
S_1	E_1	free	E_2	free
S_2	free	E_1	free	E_2

$\leq_P$

CSE Major	Slot 1	Slot 2	Slot 3	Slot 4
S_1	E_1	free	E_2 (truncate here)	
S_2	free	E_1 (truncate here)		

Poly time steps

ANY algo for stable matching problem works!

Arbitrary Y instance

Pre-process the input

Algo for X

Post-process the output

Output for Y instance

All processing is poly-time

Implications of $Y \leq_p X$

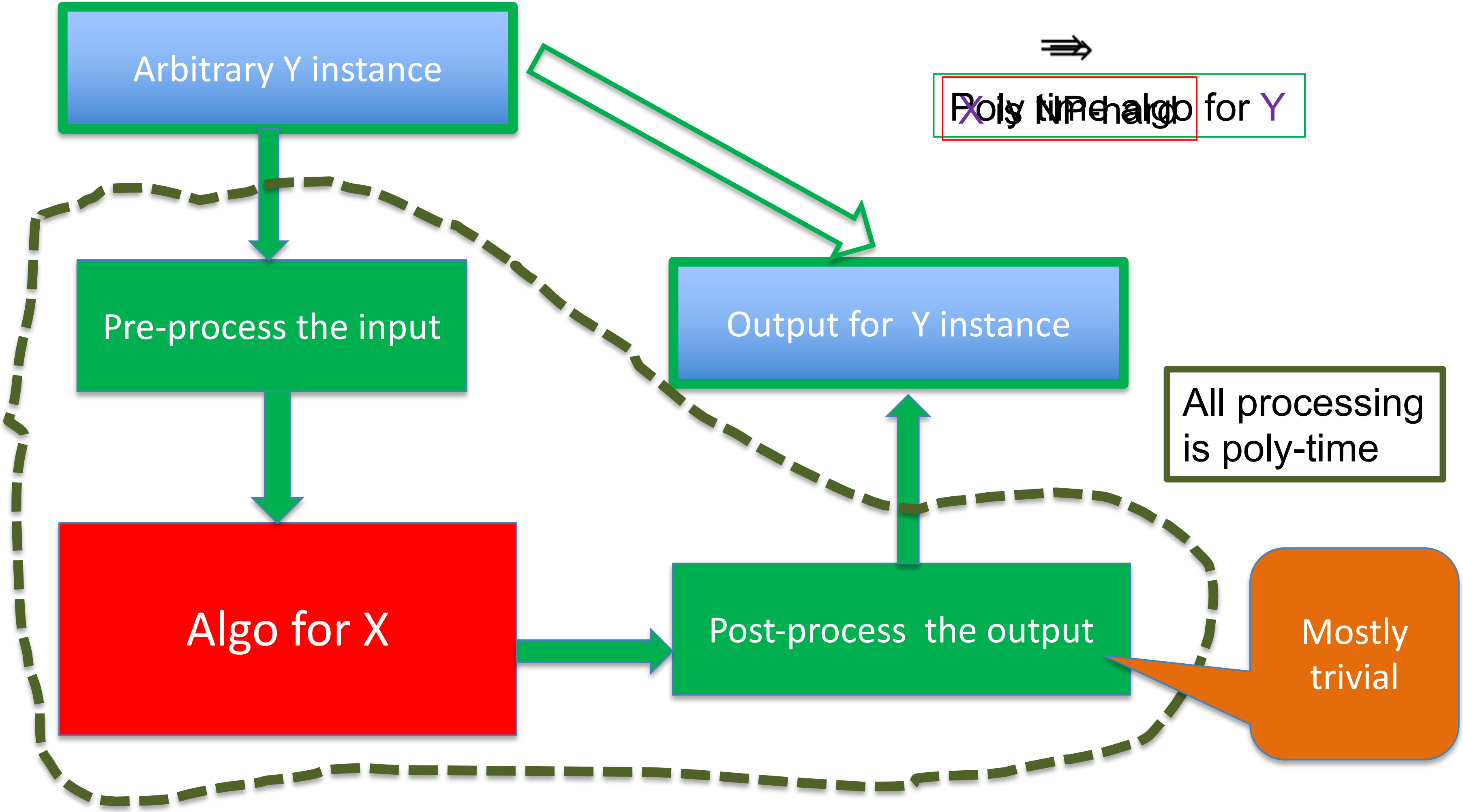
Poly time algo for X

Y is NP-hard

⟹

Poly time algo for Y

X is NP-hard



Today’s agenda

Recap of NP-completeness

NP-completeness of k-colorability