Lecture 36

CSE 331

Nov 25, 2024

Quiz 2 on Monday *after* fall break

note @328 stop following 0 views

Actions

Quiz 2 on Monday (AFTER fall break)

A gentle reminder that Quiz 2 is on **Monday, Dec 2** (the Monday *after* fall break) from **11:00-11:10am**. The lecture will start at 11:15am.

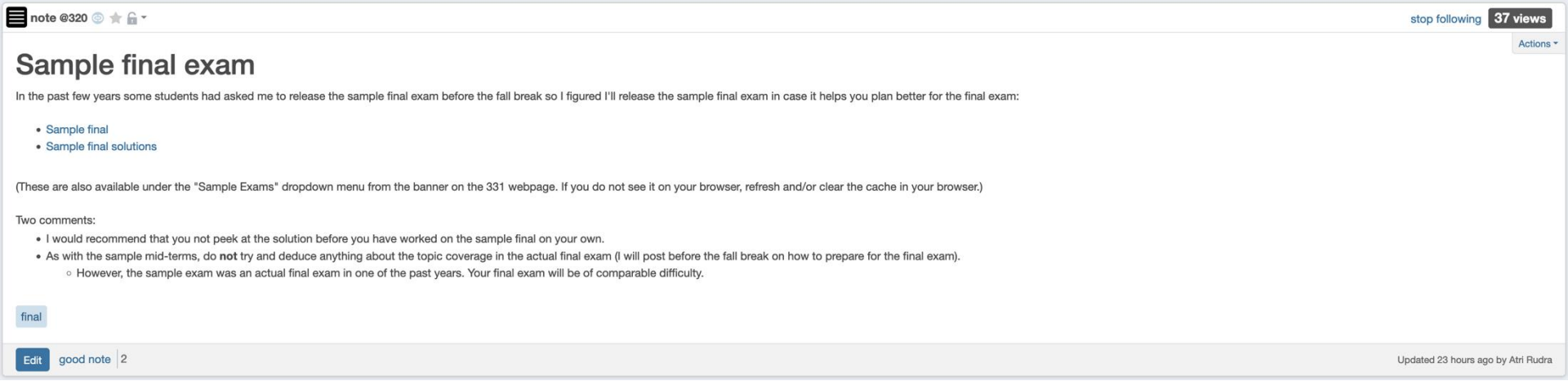
Everything until tomorrow's lecture will be on quiz 2. There will be three questions. The first two will be T/F without justification (like Q1(a) on sample final @320) and the 3rd one will be T/F with justification (like Q2(a) on sample final @320).

Like in the final exam, you are allowed two 8.5" X 11" review sheets (you can use all four sides).

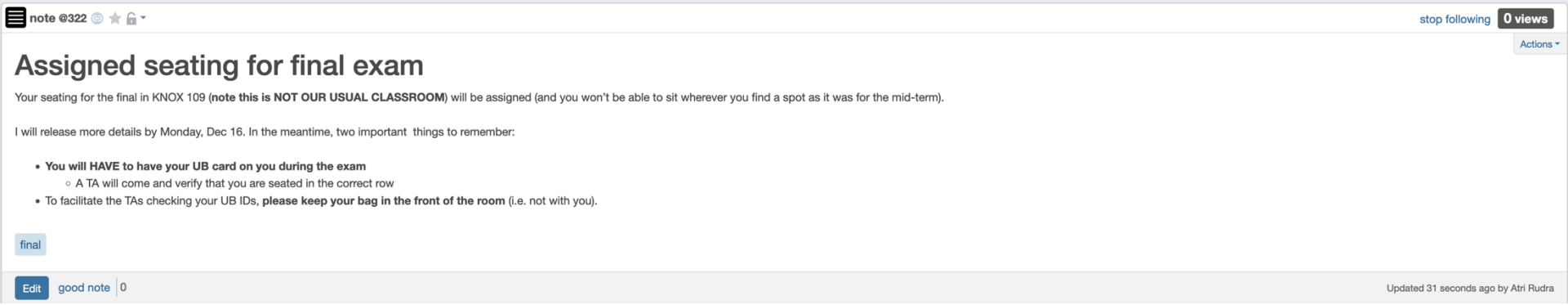
quiz2

Edit good note | 0 Updated 37 seconds ago by Atri Rudra

Sample final exam



Bring UB card to final exam



Final exam post

note @323 stop following 0 views Actions

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 NP-completeness of k-colorability (we'll finish that stuff by either by Friday, Dec 6 or Monday, Dec 9).
  - In case you want a head-start we will cover Sections 8.1-8.4 and Section 8.7 in the textbook. For the rest the schedule page details what sections of the book we have already covered.
- Exam will be from **8:30am to 11:00m** on Tuesday, **Dec 17** in **KNOX 109 (*this is NOT our usual classroom)***. Note that the exam will be for 2.5 hours and *not 3 hours* as it says on HUB.
- **DO NOT FORGET TO BRING YOUR UB CARD TO THE EXAM** (@322)

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

1. Take a look at the sample final (@320) 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 (@320).
2. 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.)
3. For the T/F questions (i.e. the first two questions), anything that was covered in class or recitations or piazza 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.
4. To get more practice for the T/F questions, review all the T/F polls on piazza (@41)
5. 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 material so y'all should definitely focus on those as well but I will not ask any "proof based" Qs on that material.
6. In previous finals, like your mid-terms, there have been questions that are either straight lifts from homeworks or are closely related and this trend will continue in the actual exam (though to a lesser extent than the mid-term). This means that you should review your homeworks (all of them) before the exam. Also make sure to review the support pages and recitation notes.
7. If you are short on time and you are prioritizing the topics to study, keep points 5 and 6 above in mind.
8. Sections in the book that were not covered at all in the class but were handed out as reading assignments or recitation notes: I can also ask any direct questions from them. In addition, it might be useful to read them to get a better feel for the material. In any case once you have read the material covered in class a couple of times, it might do your brain some good to read some different material.
9. You can bring in **two** 8.5"X11" review sheets (you can use both sides on both). Use this judiciously: they can be a very useful tool to note down some weird things you have a hard time remembering and/or noting down specific references. However, do **not** spend a lot of time preparing these sheets: they can be huge time sinks without much payoff.

Next are some suggestions for when you are **in the exam**:

1. Spend 5-10 minutes reading all of the questions in one pass: this'll let the problems germinate in your subconscious until you actually get to solving them.
2. You should have plenty of time for the exam: by my count a well prepared student should be done by spending at most one minute per point, i.e. 100 minutes. The exam will be for 150 minutes, so you will have 50 extra minutes.
3. If you are not sure how to design an algorithm for a problem in the exam I generally recommend the following sequence:
   - Try and see if you can reduce the problem to something you have already seen in class;
   - If not, then try and slightly modify an existing algorithm we have not see;
   - If not, only then try and build an algorithm from scratch.
4. Just to be sure the point above is just a recommendation-- your mileage may vary. E.g. if you immediately see a direct algorithm to solve a problem, go for it!
5. Note that even if a problem might look similar to another problem you have seen before, you *might* need to solve the new problem using a different algorithmic technique. So while "pattern matching" on the problem statement might be a good place to start be wary of surface similarities.
6. Once you reach the exam room, try to relax. Once you are there, you have done all the hard work, stressing out about the exam is not going to make the exam any easier for you. **Relax, it's just an exam!** The worst thing that can happen is you will do a bit badly: but it's just some course. I got a C in my undergrad algorithms course. So even if you do badly in 331 life will still go on and things will work out.

Apply to be a CSE 331 TA in 2025!

note @295 stop following 1 view

Actions

Want to be a UTA for 331 in 2025?

Profs. Bosse and Hayes be teaching 331 in the upcoming Spring semester and is looking for UTAs. I expect to be teaching 331 again in Fall 2025 (though this is **not** finalized and is subject to change) and will be looking for TAs then as well. So Profs. Bosse, Hayes and I are looking to jointly interviewing candidates for CSE 331 TAs for 2025 (on **zoom** tentatively the final week (Dec 18 and after) and/or the week after that (week of Dec 23, 2024).

(As an aside: I also have openings for doing research but I'll post on those once I'm done with all 331 related stuff: i.e. after the grades have been submitted.)

These will be *paid* positions. Time-commitment wise here is what we're looking for

- *Ideally,* you should be able to commit close to 10 hours/week on average. More is of course better!
- Depending on your background (e.g. if you have TAed before), we're willing to be OK with ~5 hours/week on average but no lower than that (and no more than 1-2 TAs with << 10 hrs/week).

A few important points:

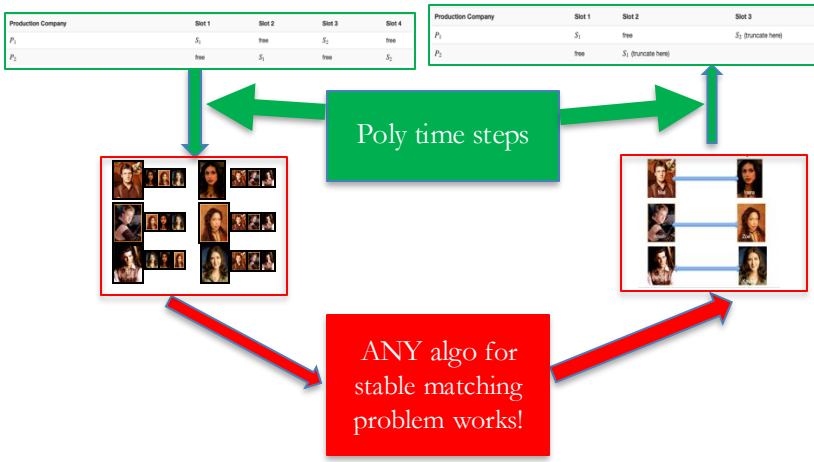
- There is *no* formal minimum grade requirement to be a 331 UTA (Of course you don't know your grade by now). For now, we're basically looking for interested students who enjoyed 331 so far and would be excited to help others.
- A large fraction of your current TAs will be TAing CSE 331 this spring (but pretty much all of them will be gone by the summer) but the SP 25 class will be about twice as large this semester so SP 25 will have many more openings (15-20) as compared to Fall 25 (10+).
- Being a 331 UTA is definitely a great experience (feel free to ask one of your TAs!) and also **a great preparation for your interviews – there is no better way to learn algorithms than to teach it!**
- The application process is basically you presenting an algorithm that is covered in class to a "mock recitation"-- once you apply, we will provide more details on the process.

Questions?

Question 2 (Syke(s) you out)

NRMP
National Resident Matching Program

$$Y \leq_P X$$



Arbitrary Y instance

Pre-process the input

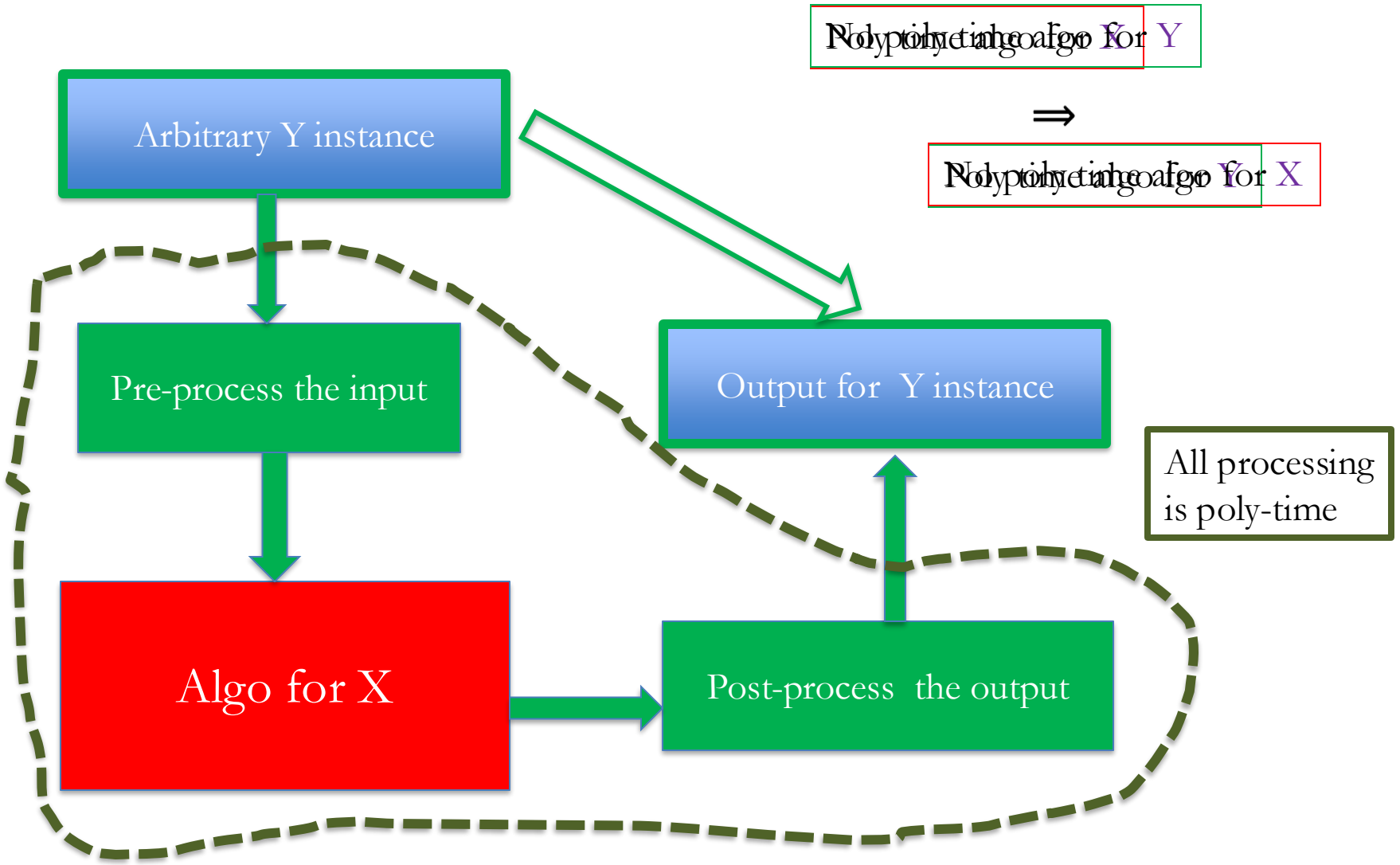
Algo for X

Post-process the output

Output for Y instance

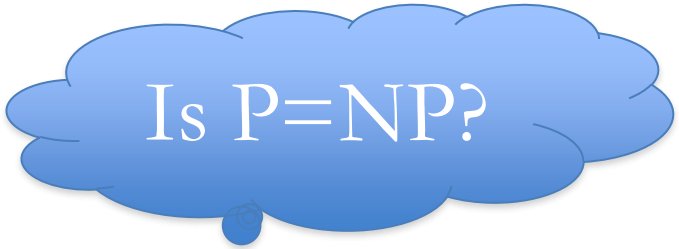
All processing is poly-time

Implications of $Y \leq_P X$



P vs NP question

P: problems that can be solved by poly time algorithms

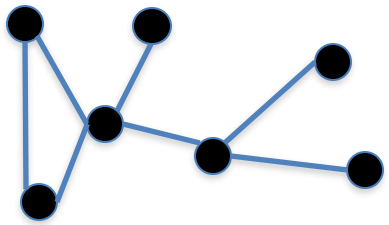


NP: problems that have polynomial time verifiable witness to optimal solution

Independent Set (IS)

Input: Graph $G = (V,E)$ and number k

Output: Yes iff G has an IS of size $\geq k$



Questions?

Plan for today

Vertex Cover (VC) problem

Define P and NP

(If there is time) define NP-completeness