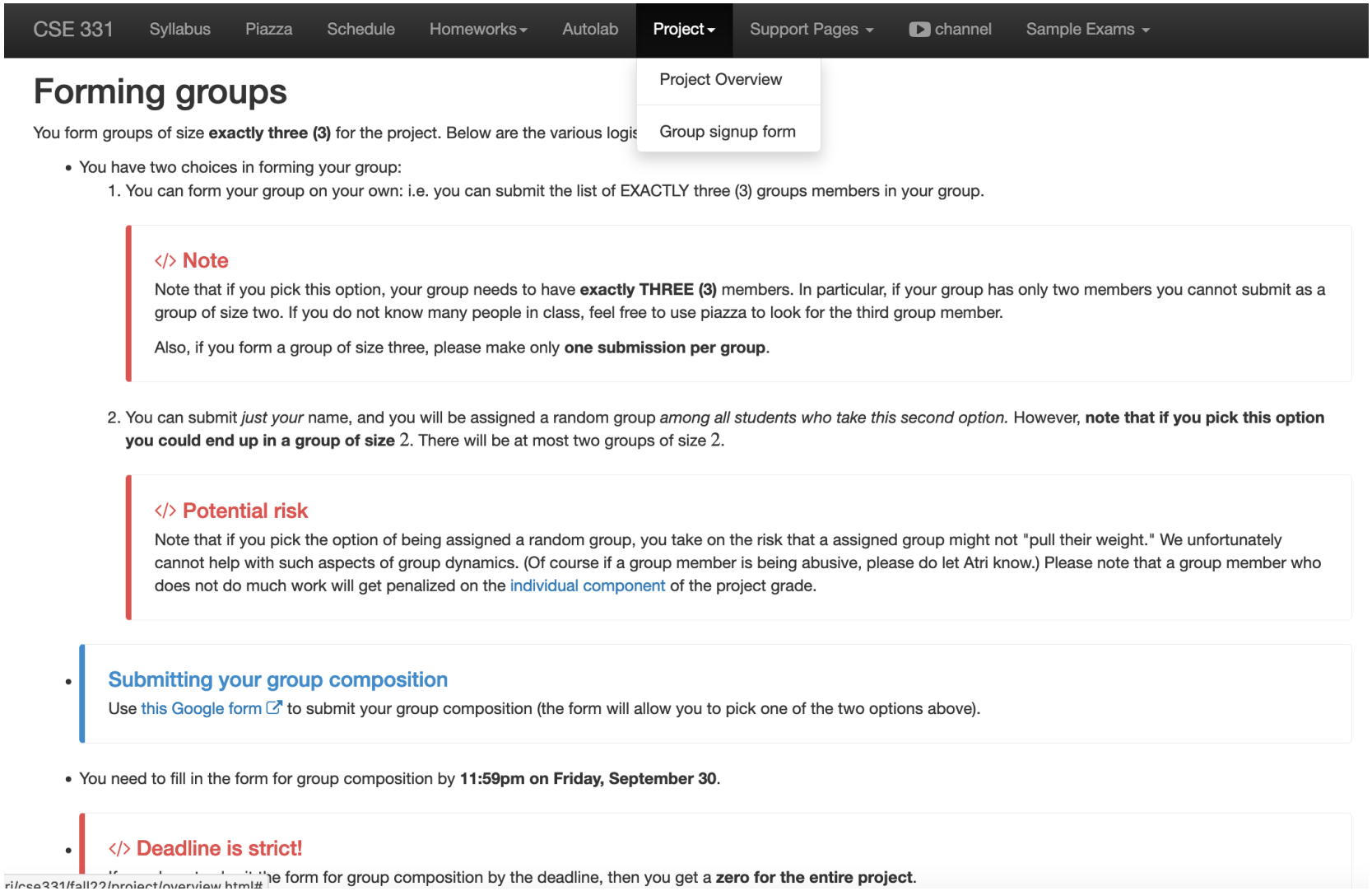
Lecture 9

CSE 331

Sep 18, 2023

Register your project groups

Deadline: Friday, Sep 29, 11:59pm



Follow ALL instructions on HW1

! Submit part (a) and (b) separately

You need to submit **two (2) PDF** files to Autolab: one for part `(a)` and one for part `(b)`. While you can assume part `(a)` as a given for part `(b)`, to get credit for part `(a)` you have to submit your solution for part `(a)` separately from part `(b)`.

Make sure you submit the correct PDF to the correct submission link on Autolab. If you do not (e.g. if you submit Q1(a) PDF to Q1(b)), then you will lose ALL points.

We recommend that you typeset your solution but we will accept scans of handwritten solution-- you have to make sure that the scan is legible.

! PDF only please

If Autolab cannot display your file, (irrespective of the reason) then you will get a zero (0) on the entire question.

Autolab might not be able to display files in formats other than PDF (e.g. Word cannot be displayed). **Note that Autolab will "accept" your submission even if you submit non-PDF file, so it is YOUR responsibility to make sure you submit in the correct format.**

Also the file size has to be **at most 3MB**.

Grading Guidelines

We will follow the usual grading guidelines for non-programming questions. Here is a high level grading rubric specific to part `(a)` of this problem:

1. `Proof idea`: `10` points.

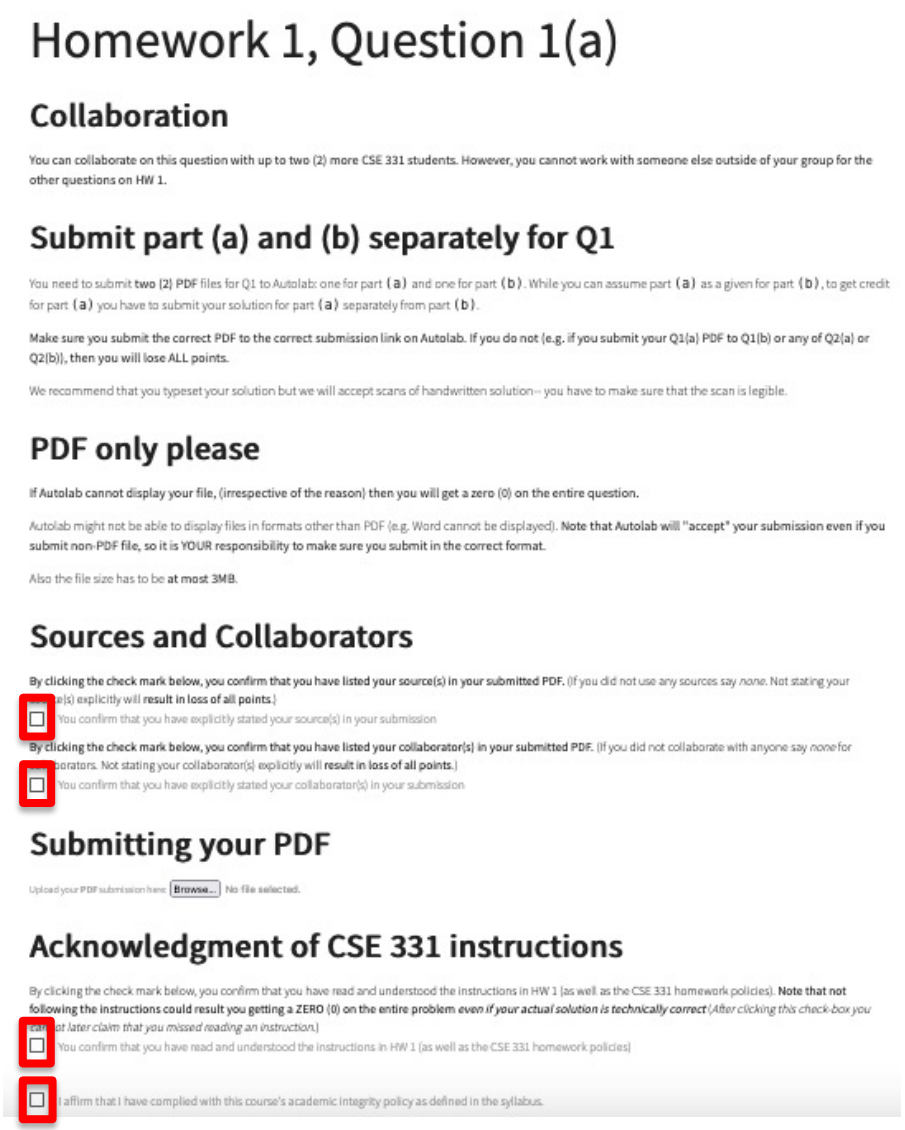
and here is the high level grading rubric for part `(b)`:

1. `Proof/Algorithm idea`: `20` points for
   If your answer is yes: the idea behind the algorithm that for any input, computes a pair of stable schedules.
   - *If your answer is no*: a counterexample idea explaining the insight behind why you think the property does not hold.
2. `Proof/Algorithm details`: `20` points for
   If your answer is yes: details of the algorithm that for any input, computes a pair of stable schedules *and* an argument as to why your algorithm will always output a pair of stable schedules for every input.
   - *If your answer is no*: a complete description of a counterexample *and* a complete proof for why the given counter example does not have any stable schedule.

! Note

If you do not have separated out proof/algorithm idea and proof/algorithm details for part `(b)`, you will get a zero (0) irrespective of the technical correctness of your solution.

They are repeated on Autolab



Homework 1, Question 1(a)

Collaboration

You can collaborate on this question with up to two (2) more CSE 331 students. However, you cannot work with someone else outside of your group for the other questions on HW 1.

Submit part (a) and (b) separately for Q1

You need to submit **two (2) PDF** files for Q1 to Autolab: one for part **(a)** and one for part **(b)**. While you can assume part **(a)** as a given for part **(b)**, to get credit for part **(a)** you have to submit your solution for part **(a)** separately from part **(b)**.

Make sure you submit the correct PDF to the correct submission link on Autolab. If you do not (e.g. if you submit your Q1(a) PDF to Q1(b) or any of Q2(a) or Q2(b)), then you will lose ALL points.

We recommend that you typeset your solution but we will accept scans of handwritten solution -- you have to make sure that the scan is legible.

PDF only please

If Autolab cannot display your file, (irrespective of the reason) then you will get a zero (0) on the entire question.

Autolab might not be able to display files in formats other than PDF (e.g. Word cannot be displayed). **Note that Autolab will "accept" your submission even if you submit non-PDF file, so it is YOUR responsibility to make sure you submit in the correct format.**

Also the file size has to be **at most 3MB**.

Sources and Collaborators

By clicking the check mark below, you confirm that you have listed your source(s) in your submitted PDF. (If you did not use any sources say *none*. Not stating your source(s) explicitly will **result in loss of all points**.)

☐ You confirm that you have explicitly stated your source(s) in your submission

By clicking the check mark below, you confirm that you have listed your collaborator(s) in your submitted PDF. (If you did not collaborate with anyone say *none* for collaborators. Not stating your collaborator(s) explicitly will **result in loss of all points**.)

☐ You confirm that you have explicitly stated your collaborator(s) in your submission

Submitting your PDF

Upload your PDF submission here: Browse... No file selected.

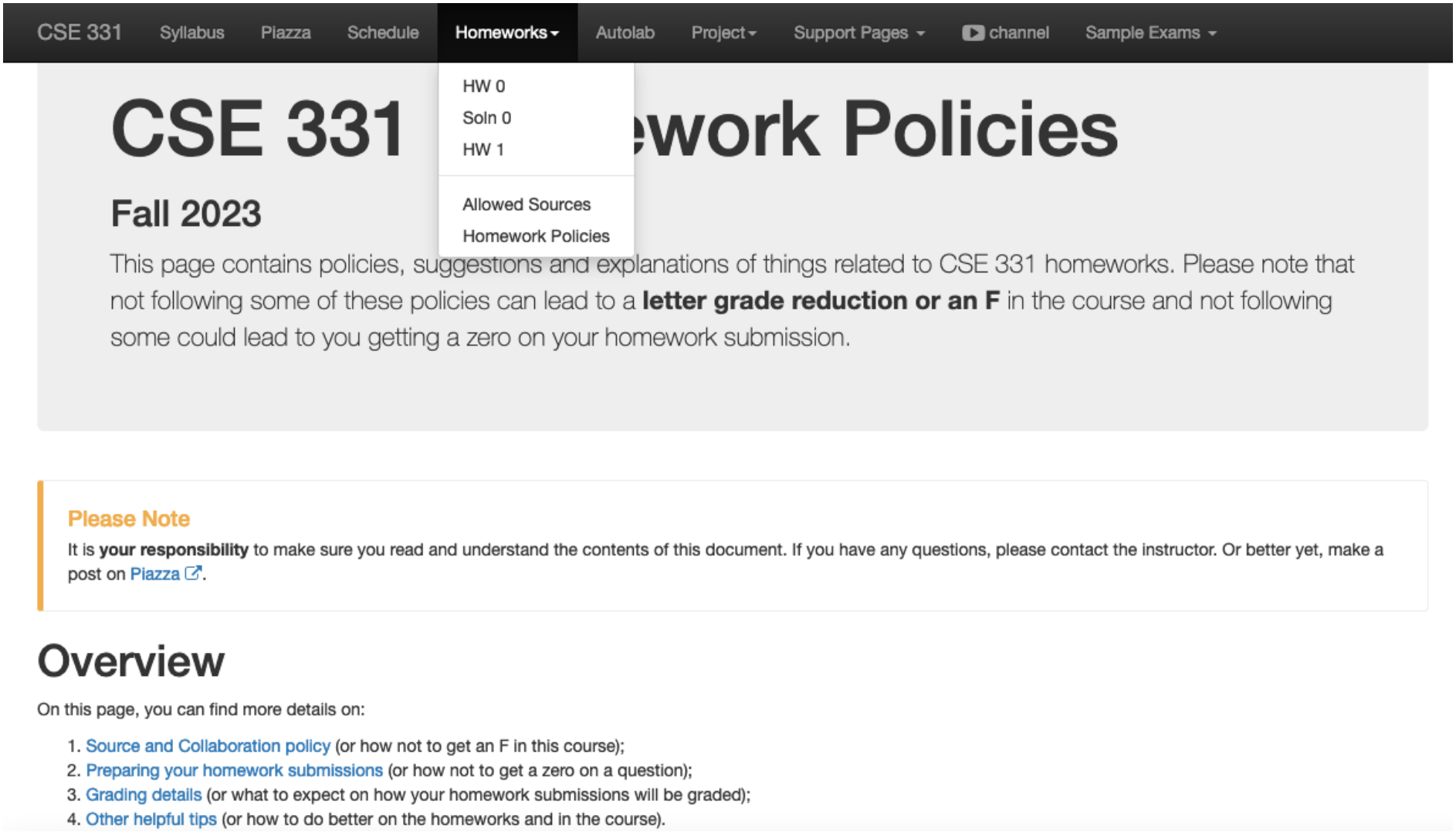
Acknowledgment of CSE 331 instructions

By clicking the check mark below, you confirm that you have read and understood the instructions in HW 1 (as well as the CSE 331 homework policies). **Note that not following the instructions could result you getting a ZERO (0) on the entire problem *even if your actual solution is technically correct*** (*After clicking this check-box you cannot later claim that you missed reading an instruction.*)

☐ You confirm that you have read and understood the instructions in HW 1 (as well as the CSE 331 homework policies)

☐ I affirm that I have complied with this course's academic integrity policy as defined in the syllabus.

Review the HW policy doc!



HW 1 (pre)post-mortem

note @97 stop following 83 views Actions

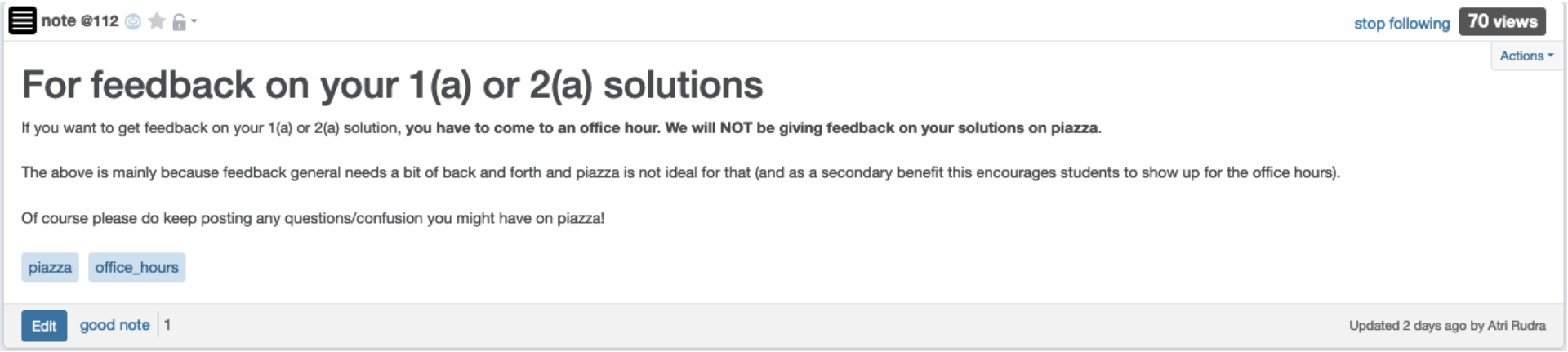
(Advance view of) post-mortem on Homework 1

The post below is from Fall 2019 that I posted *after* HW1 was due but I figured I should post it a bit earlier this time in case it is helpful to some of you as y'all work on your HW 1 submissions.

Of course this would depend pretty much on you as an individual but here are some questions, **in no particular order**, for y'all to ponder on (with some of our comments in *italics*):

- **Did you start early enough?**
  - *We recommend that you start working on the homework on Wednesday immediately after the homework is handed out itself. And distribute your hours over the week rather than wait to start till Monday (or gasp! Tuesday).*
- **Did you go to the recitations AND read the recitation notes?**
  - *Both of them help you a lot towards answering Q1(a) and Q2(a) so they are highly recommended.*
- **Did you work on the questions in correct order?**
  - *We have the current order based on what we think is most beneficial to you. In particular, we want y'all to focus more on the proof based questions, which is why they come before the programming question. But perhaps a different order would work better for you?*
- **Did you get help when you got stuck?**
  - *If you were stuck at a problem for a long time did you ask for help on piazza? Did you go to one of the office hours?*
- **Did you work on all the problems alone?**
  - *While working on all the problem by yourself will be good for you in the long run (since you are developing your proofs/algorithms skills), in the interest of time we recommend that you at least collaborate on Q2 (b).*
- **(If you submitted HW 0), did you get enough feedback?**

Feedback on Q1(a) and Q2(a)



If you need it, ask for help

Advice from TAs

CSE 331 Advice from TAs

Where students who took CSE 331 and became TAs share their experiences of how to fully utilize the class to your advantage. (And no, Atri did not pay them to say these things.)

Under Construction

This is a living document that will get updated over time. However, all the advice below is valid and you should pay attention to them!

The class is structured to your advantage

Utilize the before, during and after aspects of the course to their fullest.

Do the assigned readings before coming to class and if you get time even watch lecture videos from previous years. Atri will give you plenty of time during lecture to ask questions about the readings or the lecture itself. And of course get the most out of the assignments (Explained further below).

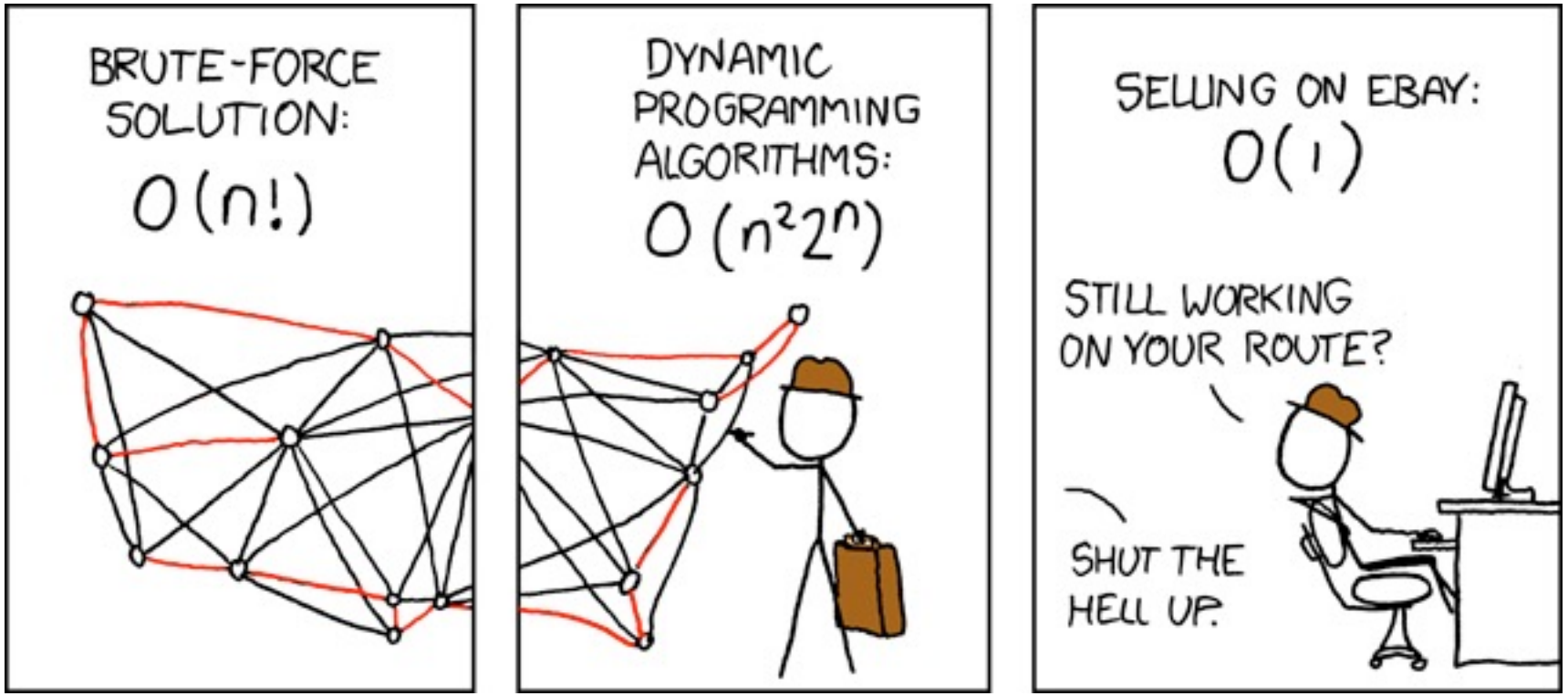
The assignments are separated into different parts for your convenience.

Questions 1 and 2

For Q1 and Q2, think of the algorithm and proof ideas as things that go inside a header (`.h`) file. They are the high level overview of how you are approaching the problem; you don't have to be very technical here. For example, listing out all the steps in your algorithm, what proof technique are you using, what property of the algorithm are you induction on, etc.

Questions/Comments?

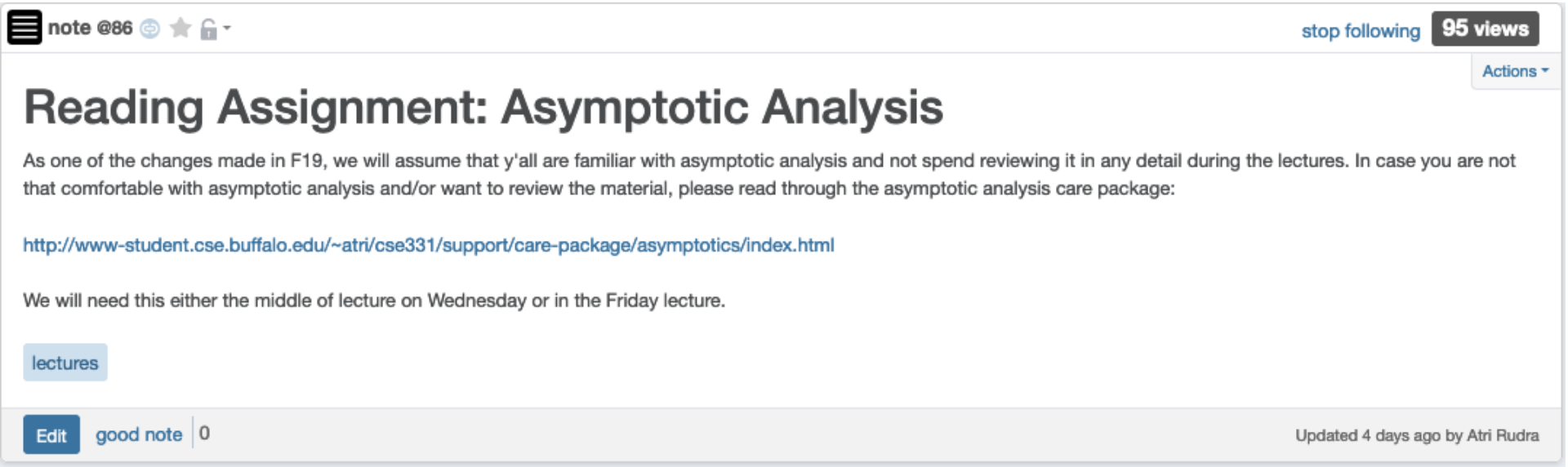
Asymptotic Analysis



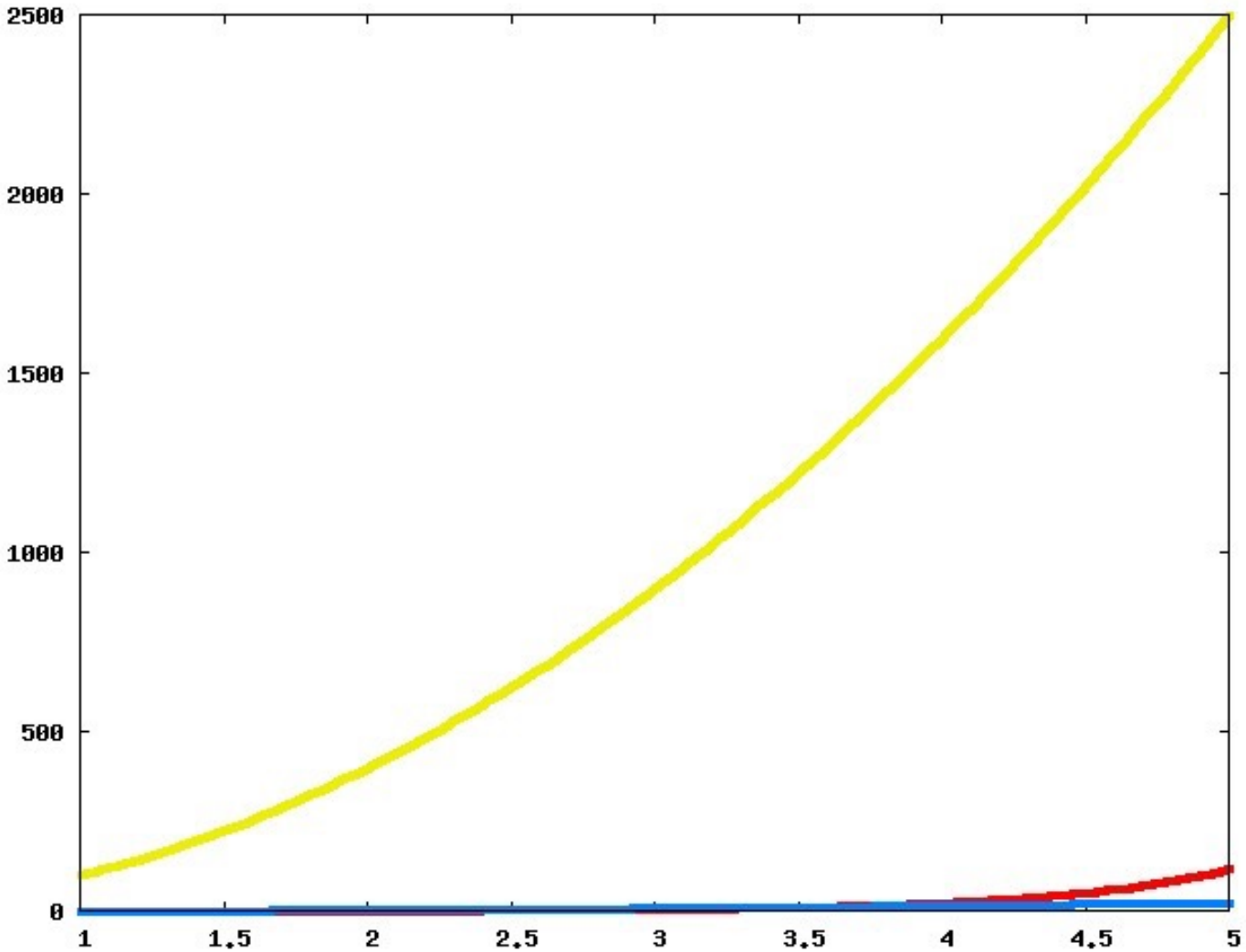
Travelling Salesman Problem

(http://xkcd.com/399/)

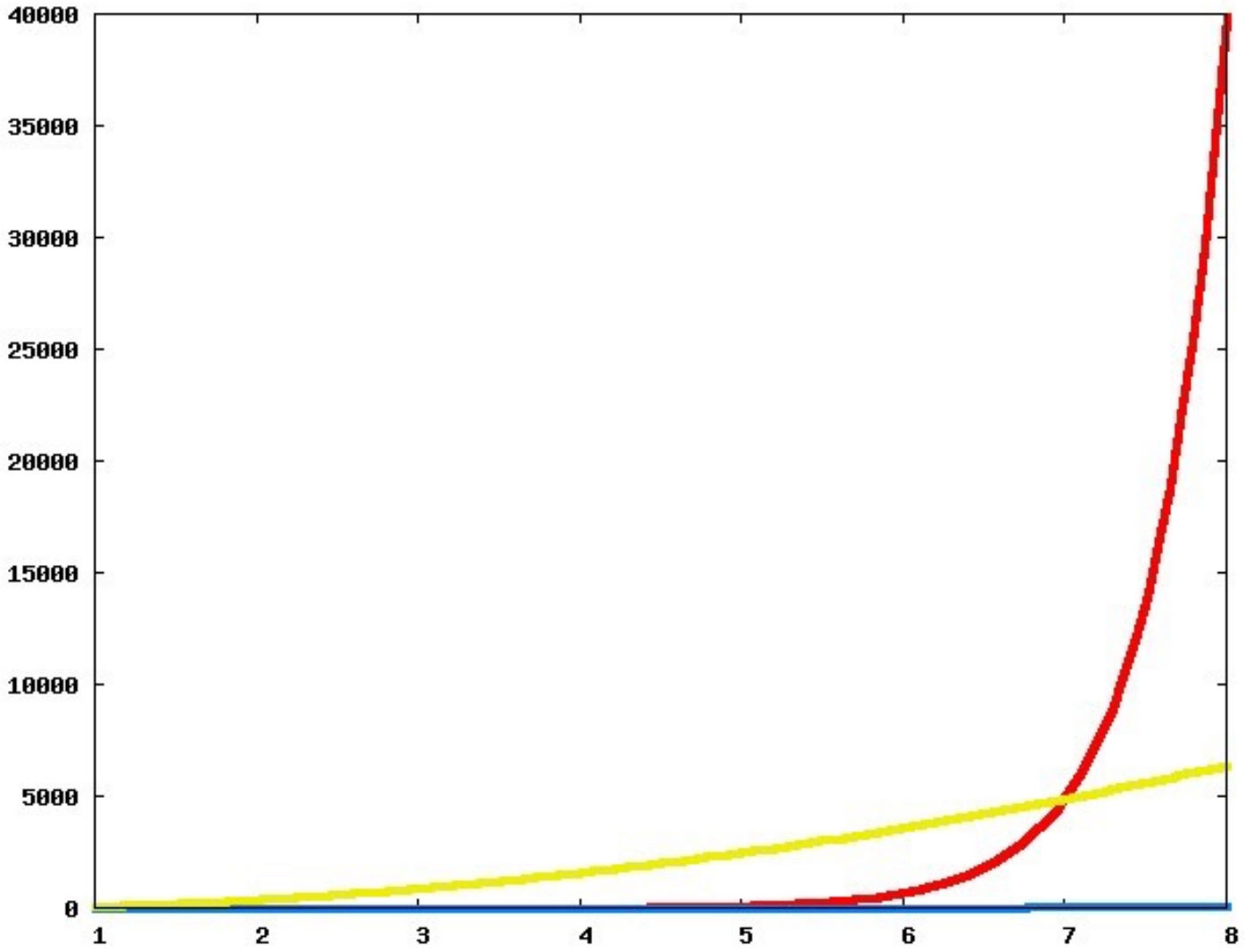
Reading Assignment for today



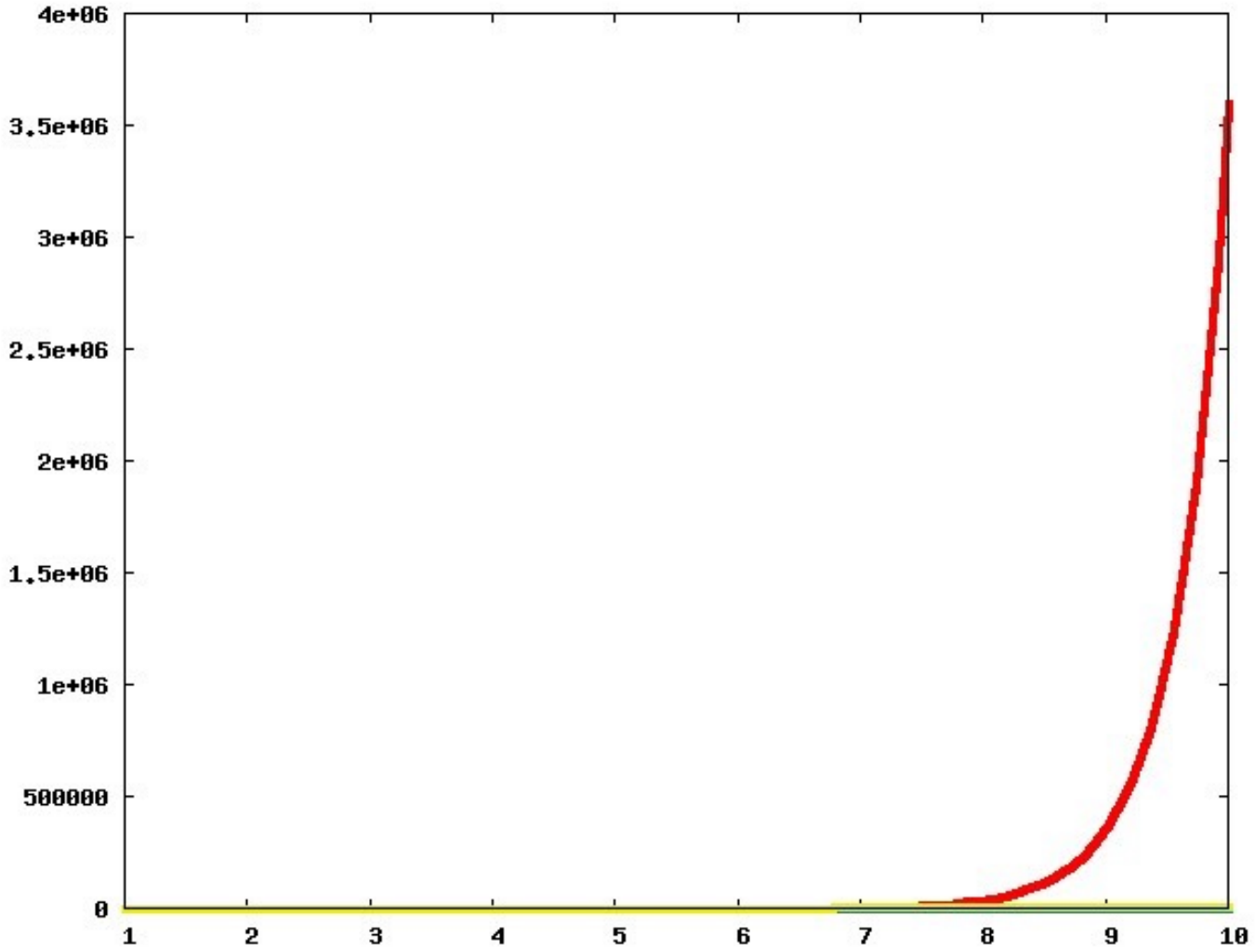
Which one is better?



Now?



And now?

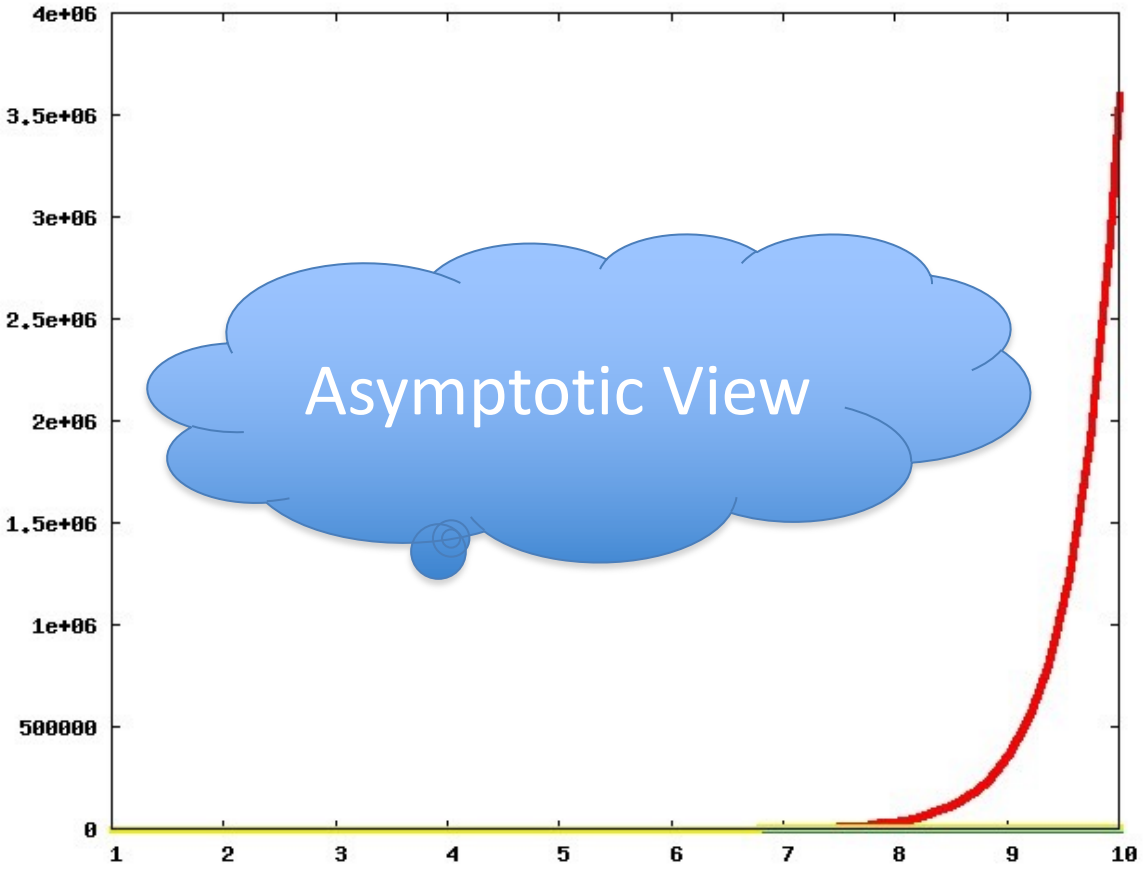


The actual run times

n!

$100n^2$

n^2



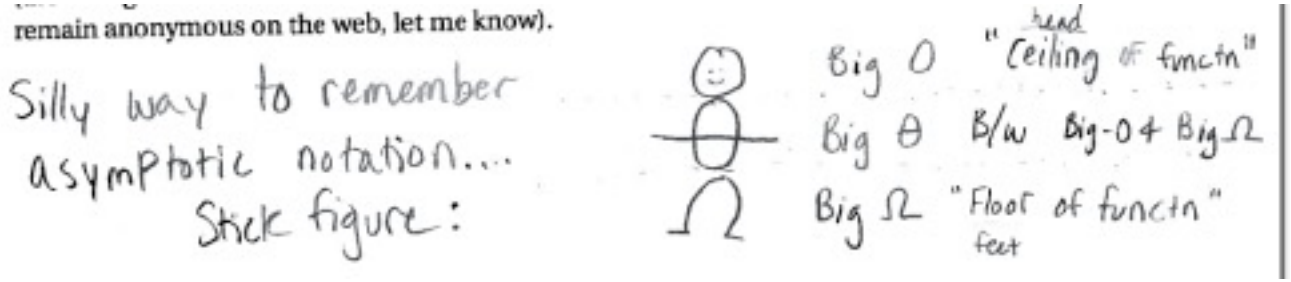
Asymptotic Notation

≤ is O with glasses

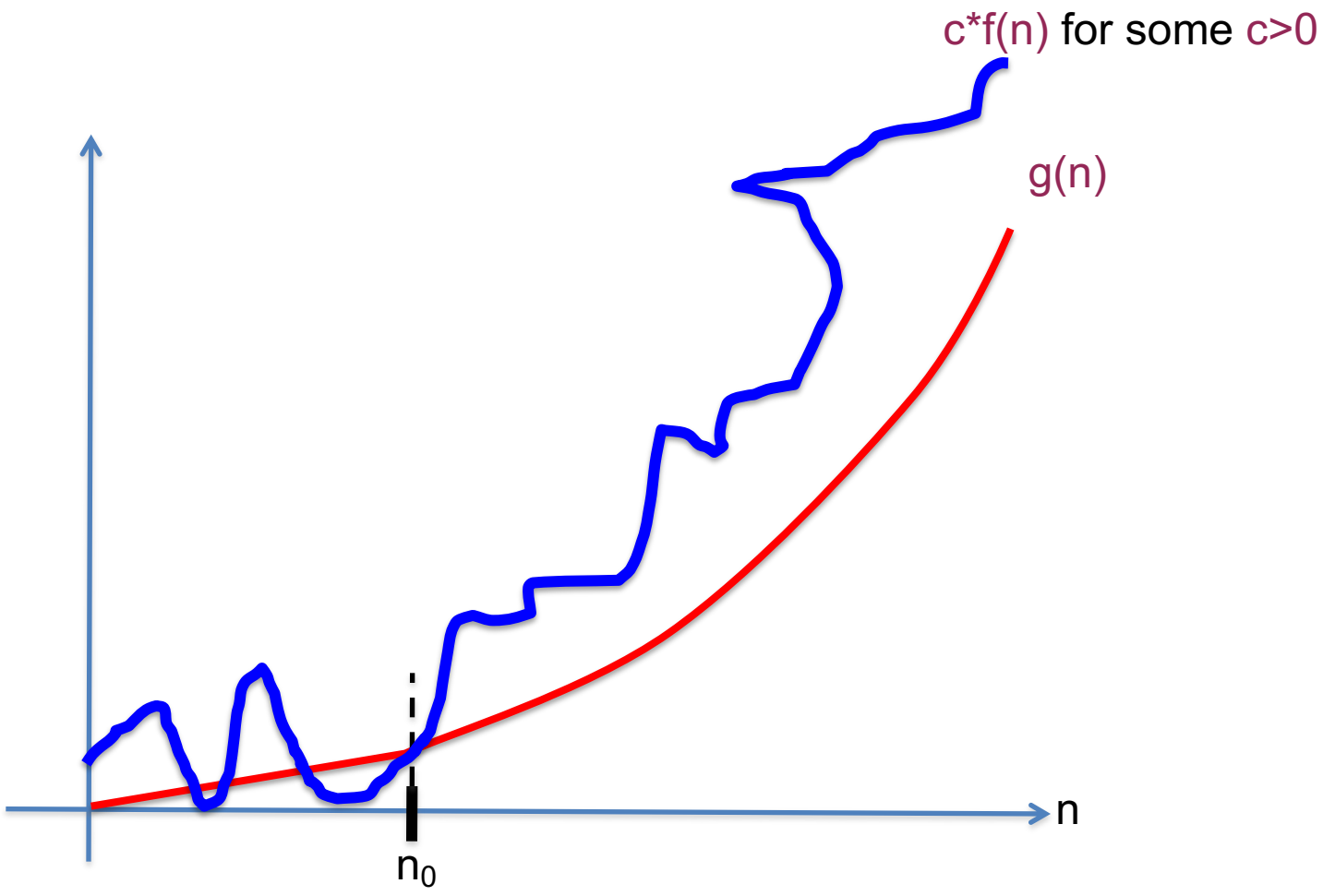
≥ is Ω with glasses

= is Θ with glasses

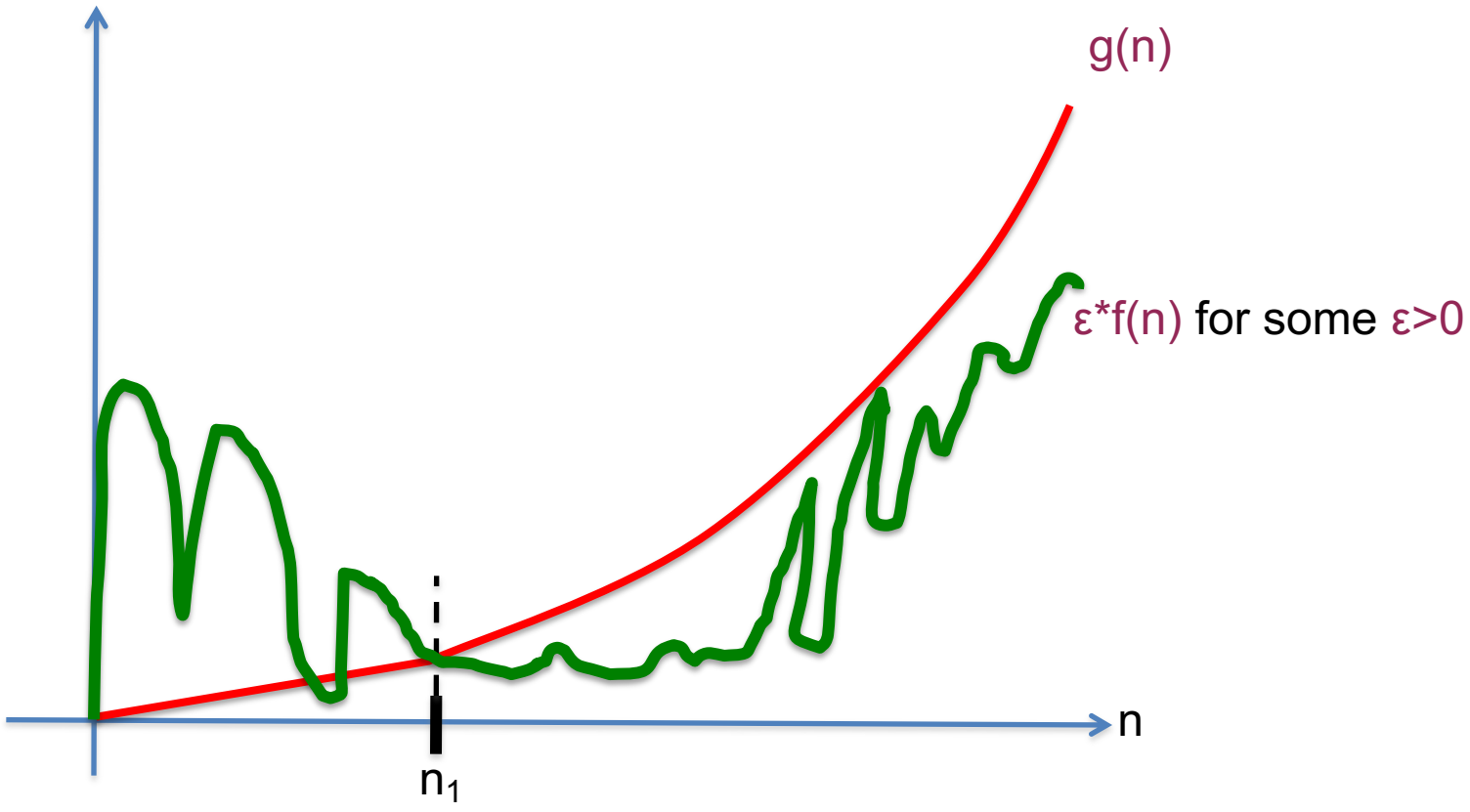
Another view



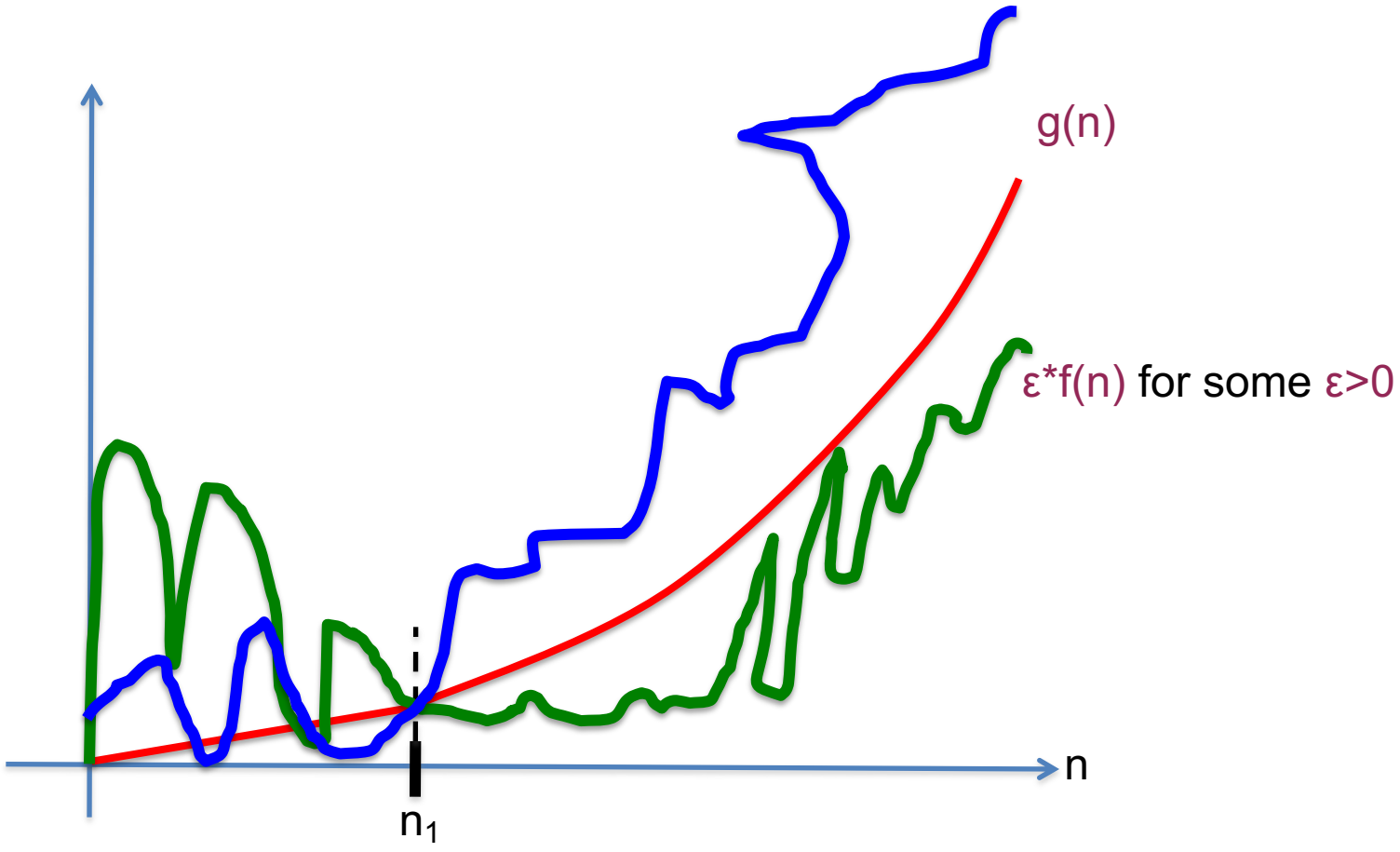
g(n) is O(f(n))



g(n) is Ω(f(n))



g(n) is Θ(f(n))



Properties of O (and Ω)

Transitive

g is O(f) and f is O(h) then
g is O(h)

Step 1 // O(n) time
Step 2 // O(n) time

Additive

g is O(h) and f is O(h) then
g+f is O(h)

Overall:
O(n) time

Multiplicative

g is $O(h_1)$ and f is $O(h_2)$ then
$g*f$ is $O(h_1*h_2)$

Overall:
$O(n^2)$ time

While (loop condition) // $O(n^2)$ iterations
Stuff happens // O(1) time

Questions?

Rest of today's agenda

$O(n^2)$ implementation of the Gale-Shapley algorithm

Some practice with run time analysis

Another Reading Assignment

CSE 331 Support Pages

Analyzing the worst-case runtime of an algorithm

Some notes on strategies to prove Big-Oh and Big-Omega bounds on runtime of an algorithm.

The setup

Let $\mathcal{A}$ be the algorithm we are trying to analyze. Then we will define $T(N)$ to be the worst-case run-time of $\mathcal{A}$ over all inputs of size N. Slightly more formally, let $t_{\mathcal{A}}(\mathbf{x})$ be the number of steps taken by the algorithm $\mathcal{A}$ on input $\mathbf{x}$. Then

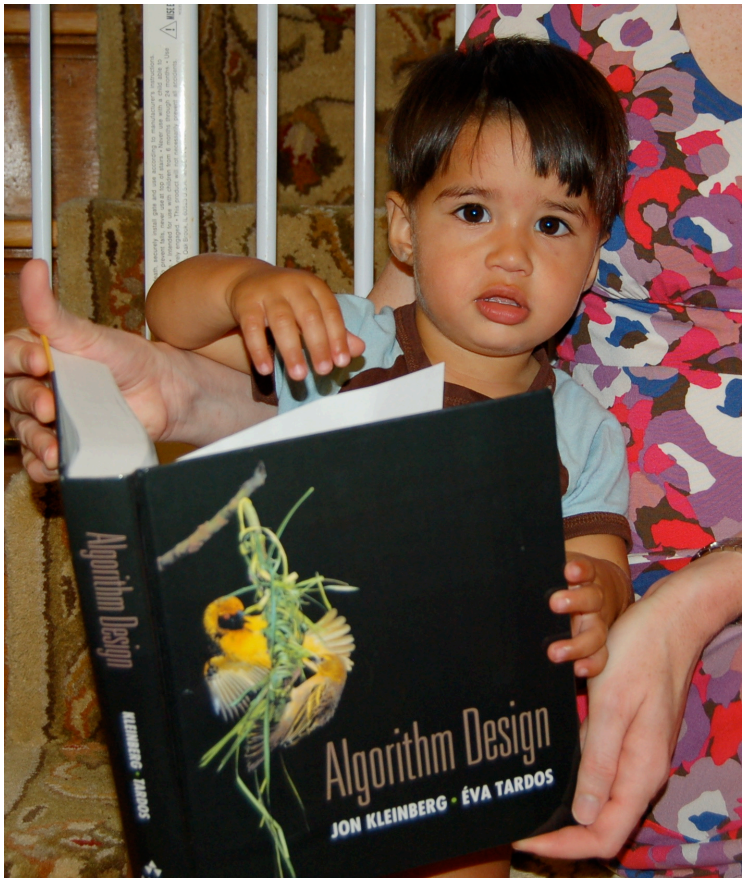
$$T(N) = \max_{\mathbf{x}:\mathbf{x} \text{ is of size } N} t_{\mathcal{A}}(\mathbf{x}).$$

In this note, we present two useful strategies to prove statements like $T(N)$ is $O(g(N))$ or $T(N)$ is $\Omega(h(N))$. Then we will analyze the run time of a very simple algorithm.

Preliminaries

We now collect two properties of asymptotic notation that we will need in this note (we saw these in class today).

Reading Assignments



Sections 1.1, 1.2, 2.1, 2.2 and 2.4 in [KT]

Gale-Shapley Algorithm

Intially all men and women are free

While there exists a free woman who can propose

At most n^2 iterations

Let w be such a woman and m be the best man she has not proposed to

w proposes to m

If m is free

(m,w) get engaged

Else (m,w') are engaged

If m prefers w' to w

w remains free

Else

(m,w) get engaged and w' is free

O(1) time implementation

Output the engaged pairs as the final output

Implementation Steps

(0) How to represent the input?

(1) How do we find a free woman w?

(2) How would w pick her best unproposed man m?

(3) How do we know who m is engaged to?

(4) How do we decide if m prefers w’ to w?

Overall running time

Init(1-4)

n^2 X (Query/Update(1-4))

Questions?

Rest on the board…