Lecture 8

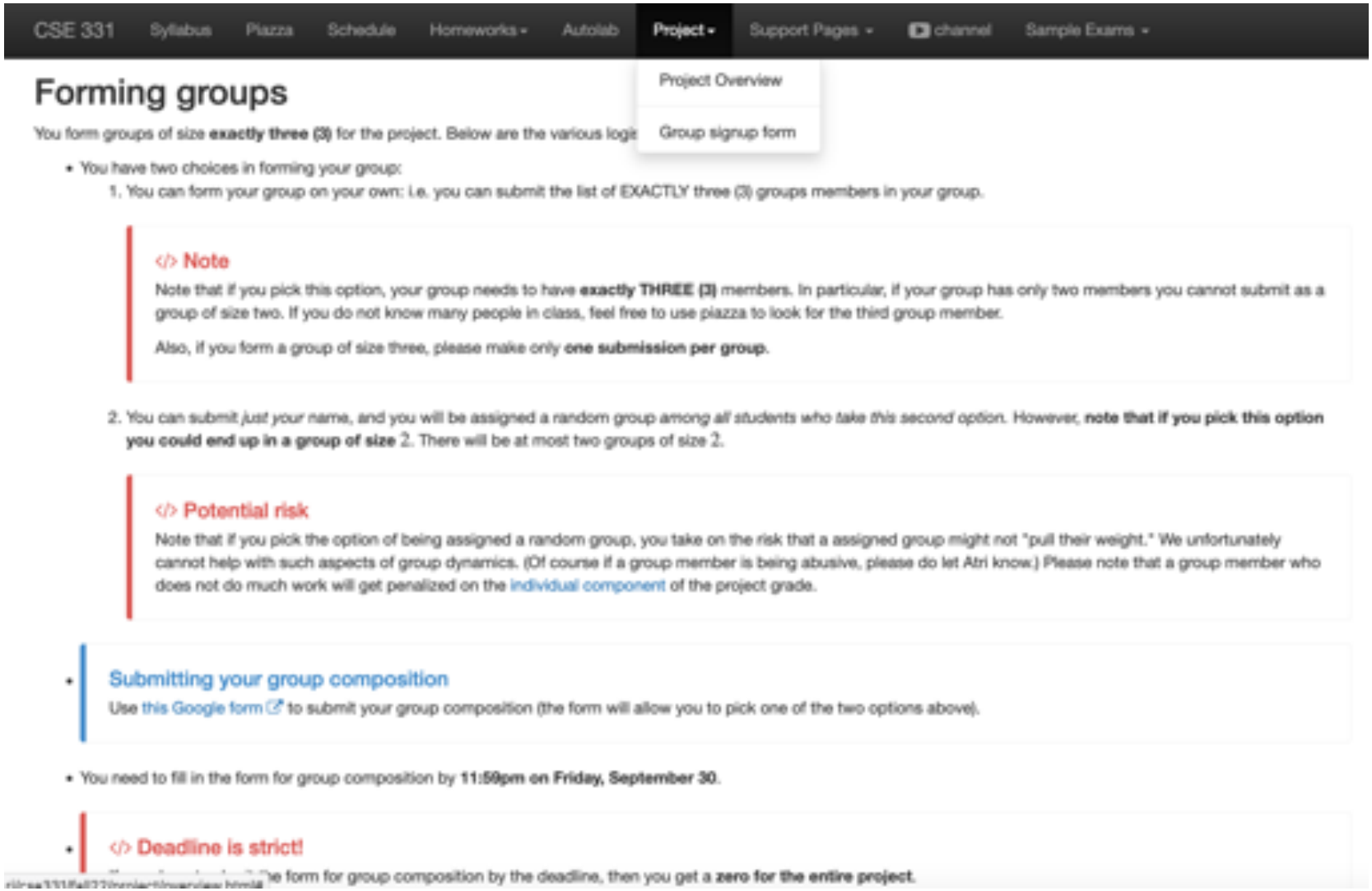
CSE 331

Sep 16, 2022

If you need it, ask for help

Register your project groups

Deadline: Friday, Sep 30, 11:59pm



Your UBIT ID is

xyz if your email ID is xyz@buffalo.edu

NOT

xyz@buffalo.edu

Your UB person number

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 you solution for part `(a)` separately from part `(b)`.

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

Autolab might not be able to display files in formats other than PDF (e.g. Word cannot be displayed). **If Autolab cannot display your file, then you will get a zero (0) on the entire question. 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.** However, after submission, Autolab will try and display your non-PDF submission and it should give an error message then. 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 idea`: `20` points for a counterexample idea explaining the insight behind why you think the property does not hold.
2. `Proof details`: `20` points for 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 idea and proof details for part `(b)`, you will get a zero (0) irrespective of the technical correctness of your solution..

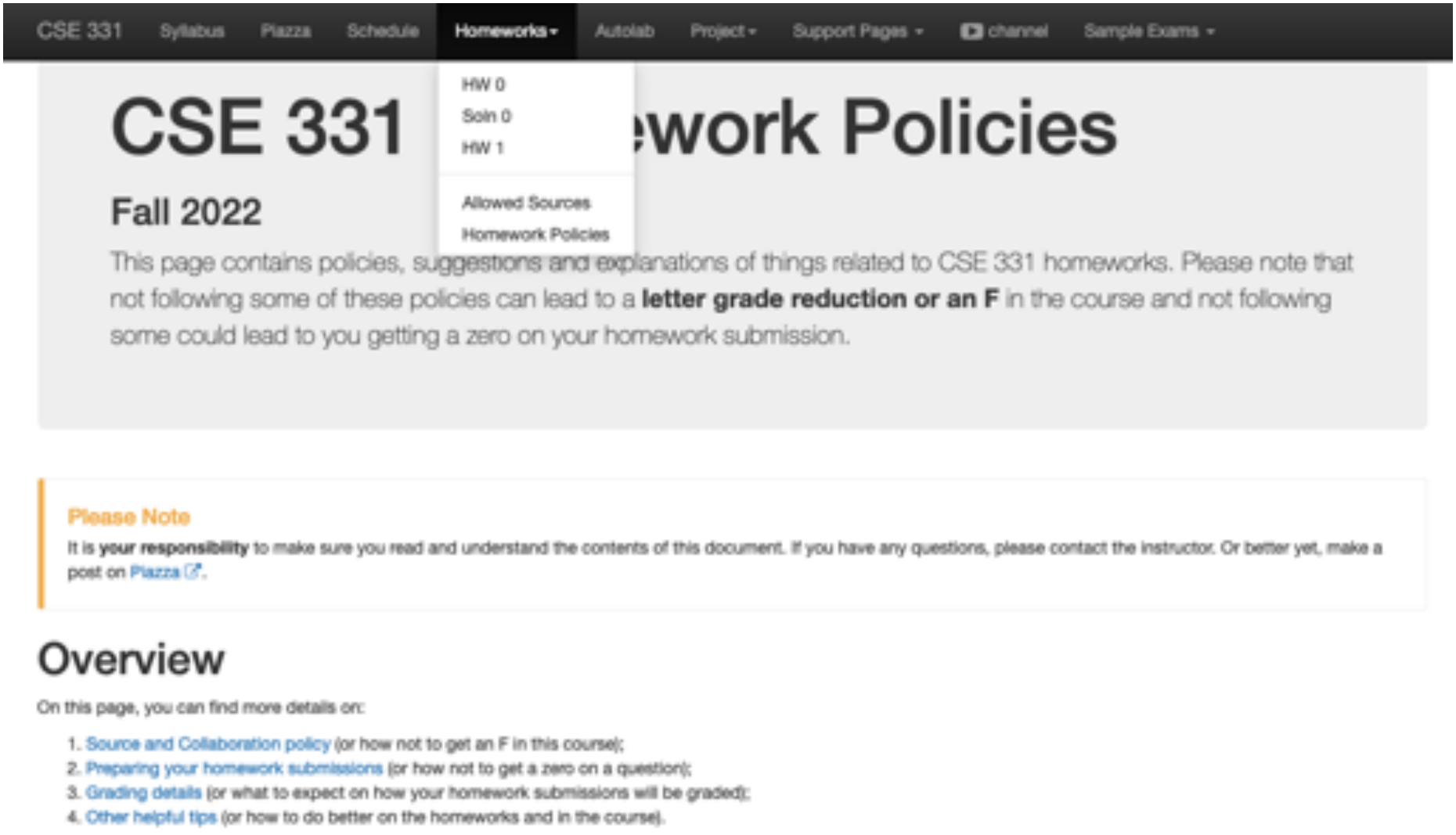
Templates

Download LaTeX template. | Download Microsoft Word template.

! Note

You must explicitly list your sources and collaborators when you upload your submission to Autolab. Note that there are only five allowed sources. If you have used a source that is not allowed, please do **not** submit your homework. If you did not consult any source or did not collaborate with anyone just say `None`.

Review the HW policy doc!



HW 1 (pre)post-mortem

note dog

stop following 1 view

Actions

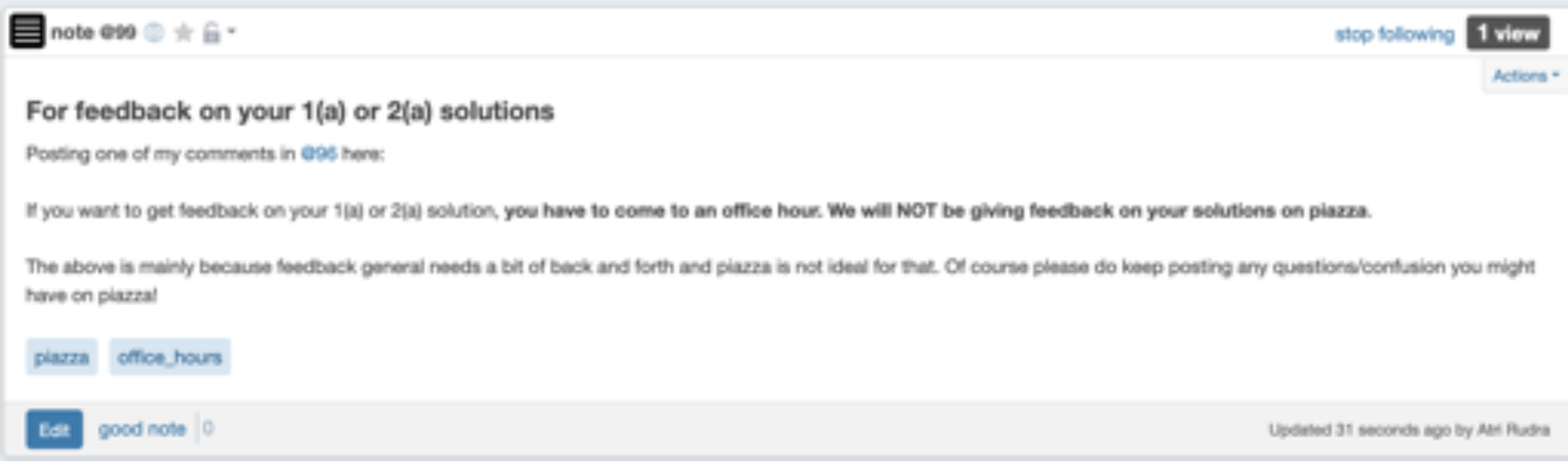
(Advance view of) post-mortem on Homework 1

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?**
  - *(This is going to be true for all homeworks so extrapolate this advice for future homeworks.) If you lost points, did you understand why you lost points? If not, did you go talk with the TA who graded your submission to ask why?*
    - *If you did understand why you lost points, did you figure out how you could have changed your thought process (and hence your solution) to get a level 0? If not, did you talk with a TA to get their thoughts on how they would change your solution to make it correct?*
- **Did you go an office hours early enough?**
  - *We recommend that you think about a problem early enough so that you can go to an office hours before Monday, when the office hours have relatively low traffic.*
- **Did you leave yourself enough time to write down your proofs?**
  - *Even if you have the right idea(s) in your heads, it takes time to write them down in a manner that would be understandable to the TA (who will not have the benefit of having access to your thoughts beyond what you have written down). We recommend that you start early to write down your solutions and definitely not wait till the end to write them down.*
- **Did you submit early (and often) enough?**
  - *We recommend that you submit early enough (and e.g. not very late at night when you might be prone to making silly mistakes) so that you do not make mistakes such as submitting in word instead of PDF or submitting your part (a) solution to part (b). In particular, we recommend that you submit as soon as you have a reasonable version of your solution written up. You can re-submit a better version later on (we only grade the last version that is submitted).*
- **Were there specific background material that you struggled with?**
  - *If so, review the relevant material from the support pages.*
- **Did you ask the right questions in the office hours?**

Feedback on 1(a)/2(a) solutions



Questions/Comments?

Gale-Shapley Algorithm

Intially all men and women are free

While there exists a free woman who can propose

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

Output the engaged pairs as the final output

The Lemmas

Lemma 1: The GS algorithm has at most n^2 iterations

Lemma 2: S is a perfect matching

Lemma 3: S has no instability

Proof Details of Lemma 1

Gale Shapley algorithm terminates

This page collects material from Fall 17 incarnation of CSE 331, where we proof details for the claim that the Gale-Shapley algorithm terminates in $O(n^2)$ iterations.

Where does the textbook talk about this?

Section 1.1 in the textbook has the argument (though not in as much detail as below).

Fall 2017 material

Here is the lecture video (it starts from the part where we d the proof details):

Questions/Comments?

Proof technique de jour

Proof by contradiction

Assume the negation of what you want to prove

After some reasoning

Source: 4simpsons.wordpress.com

Two obervations

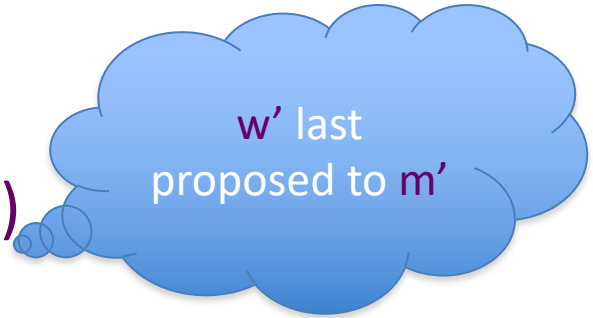
Obs 1: Once m is engaged he keeps getting engaged to "better" women

Obs 2: If w proposes to m' first and then to m (or never proposes to m) then she prefers m' to m

Proof of Lemma 3

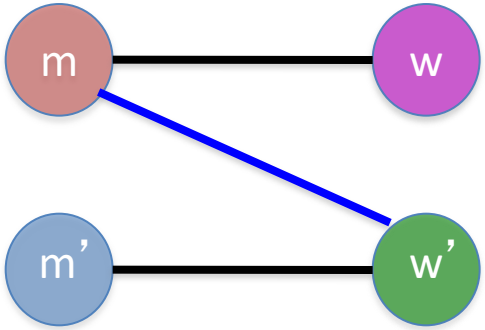
By contradiction

Assume there is an instability (m,w')



m prefers w' to w

w' prefers m to m'



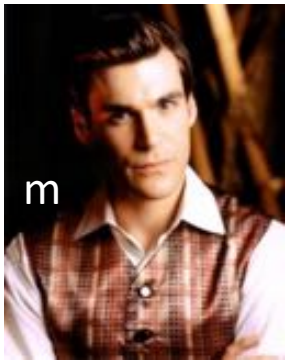
Contradiction by Case Analysis

Depending on whether w’ had proposed to m or not

Case 1: w’ never proposed to m

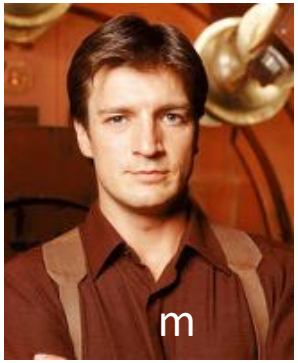
w’ prefers m’ to m — By Obs 2

Assumed w’ prefers m to m’



Source: 4simpsons.wordpress.com

Case 2: w’ had proposed to m



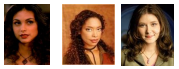
Case 2.1: m had accepted w’ proposal

m is finally engaged to w

Thus, m prefers w to w’

4simpsons.wordpress.com

By Obs 1



Case 2.2: m had rejected w’ proposal

m was engaged to w’’ (prefers w’’ to w’)

By Algo def

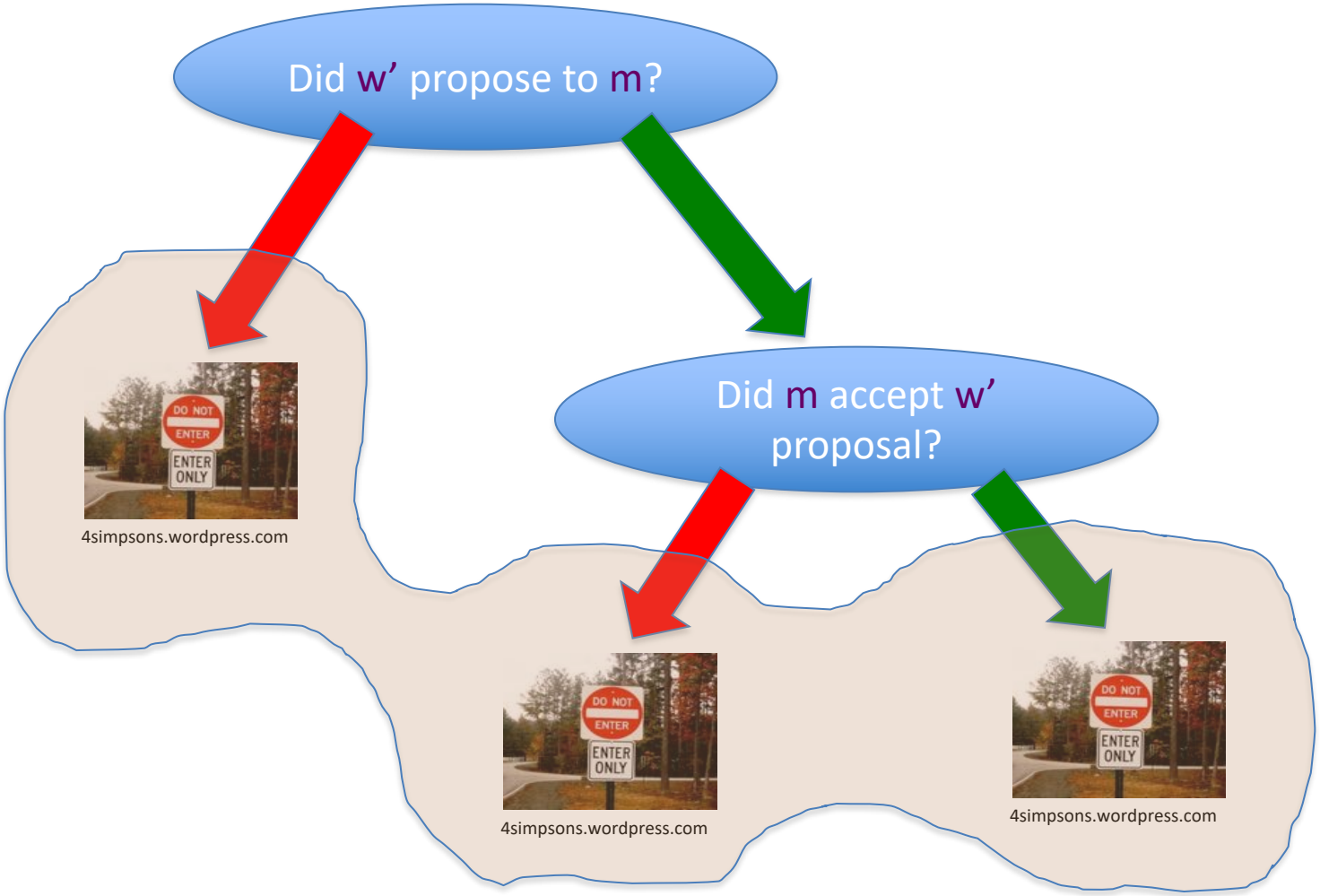
m is finally engaged to w (prefers w to w’’)

By Obs 1

m prefers w to w’

4simpsons.wordpress.com

Overall structure of case analysis



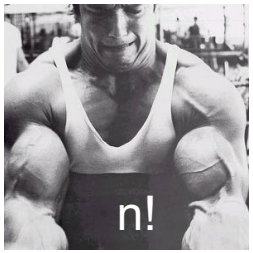
Questions?

Extensions

Fairness of the GS algorithm

Different executions of the GS algorithm

Main Steps in Algorithm Design



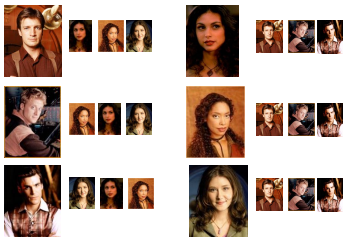
Problem Statement

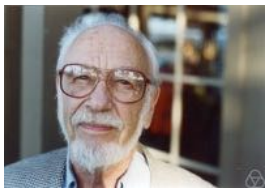
Problem Definition

Algorithm

"Implementation"

Analysis





Correctness Analysis

Definition of Efficiency

An algorithm is efficient if, when implemented, it runs quickly on real instances

Implemented where?

Platform independent definition

What are real instances?

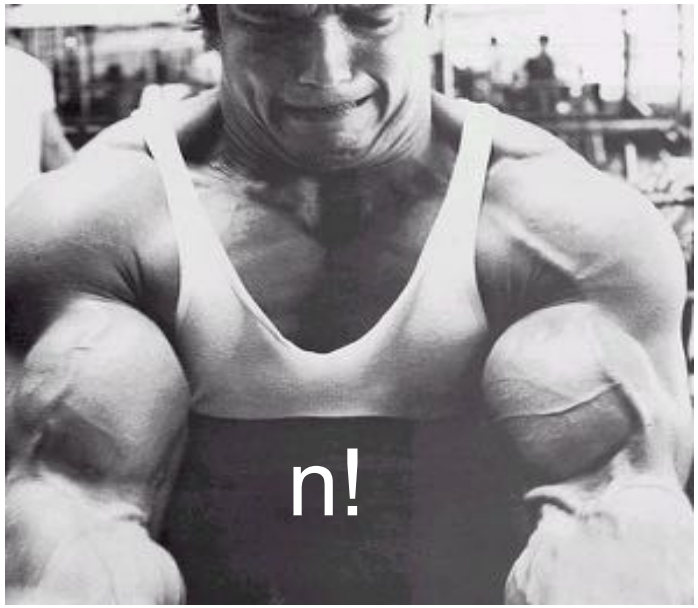
Worst-case Inputs

Efficient in terms of what?

$N = 2n^2$ for SMP

Input size N

Definition-II



Analytically better than brute force

How much better? By a factor of 2?

Definition-III

Should scale with input size

If N increases by a constant factor, so should the measure

Polynomial running time

At most $c \cdot N^d$ steps ($c>0$, $d>0$ absolute constants)

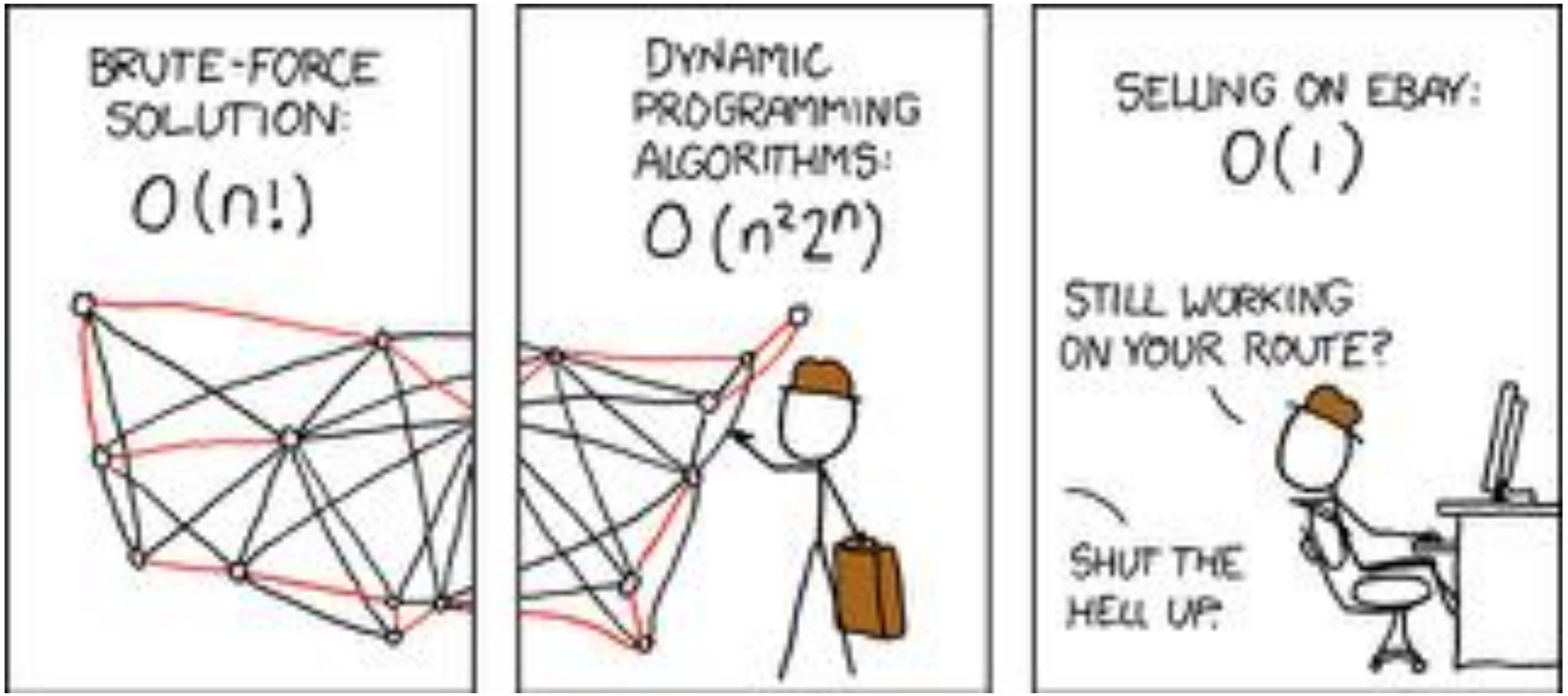
Step: “primitive computational step”

More on polynomial time

Problem centric tractability

Can talk about problems that are not efficient!

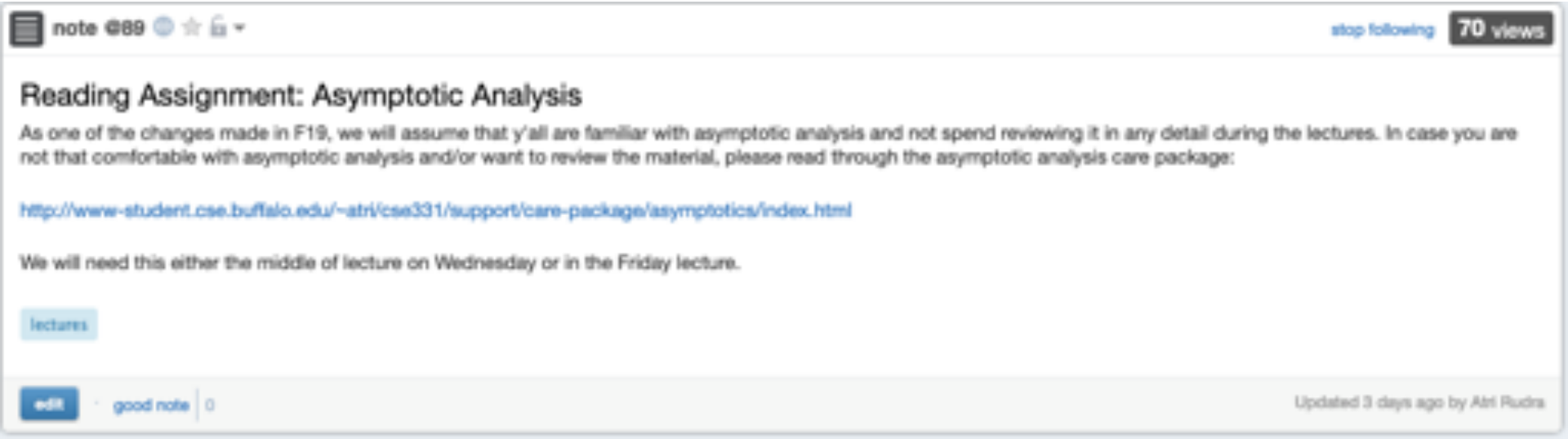
Asymptotic Analysis



Travelling Salesman Problem

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

Reading Assignment for today


note @89

stop following 70 views

Reading Assignment: Asymptotic Analysis

As one of the changes made in F19, we will assume that y'all are familiar with asymptotic analysis and not spend reviewing it in any detail during the lectures. In case you are not that comfortable with asymptotic analysis and/or want to review the material, please read through the asymptotic analysis care package:

http://www-student.cse.buffalo.edu/~atri/cse331/support/care-package/asymptotics/index.html

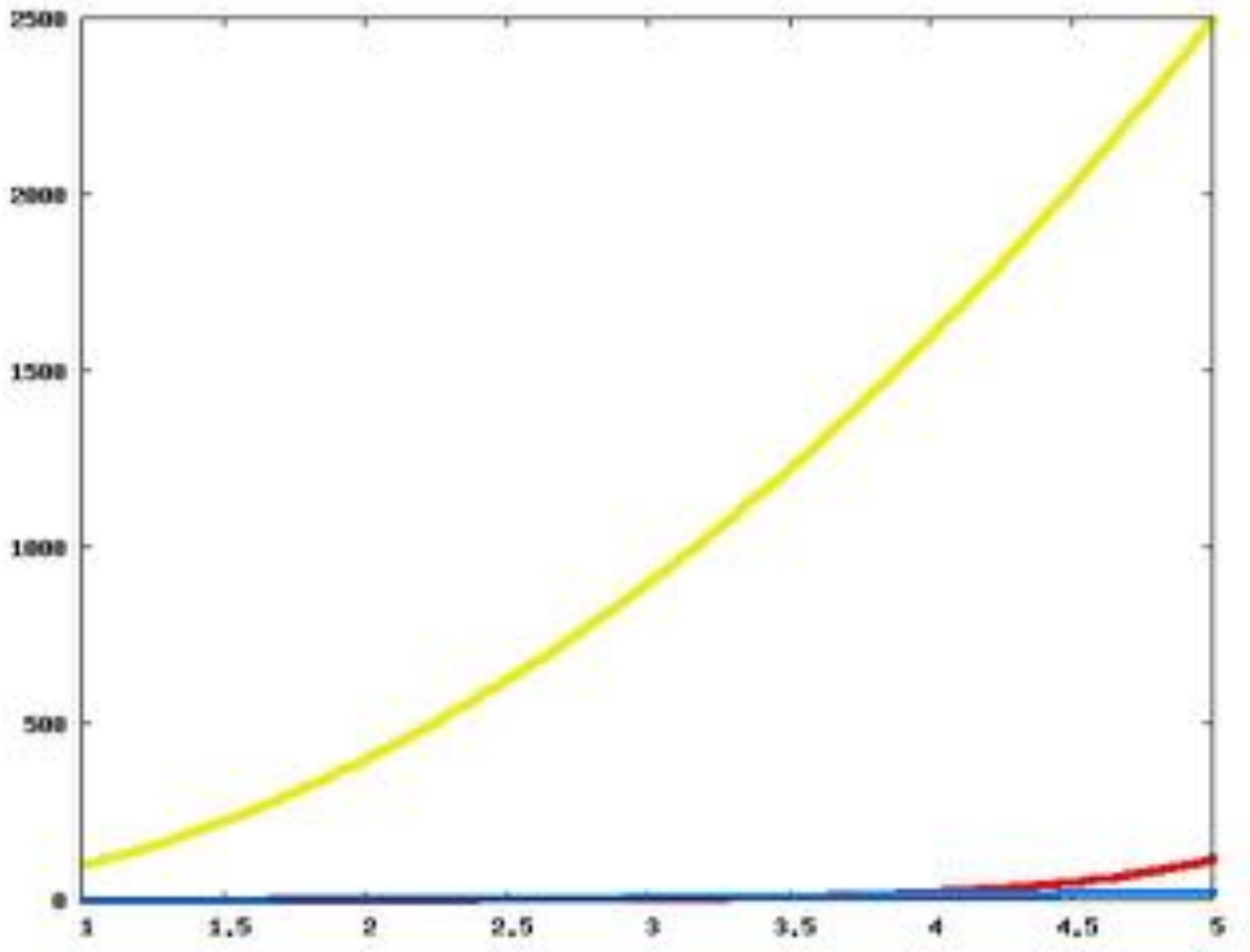
We will need this either the middle of lecture on Wednesday or in the Friday lecture.

lectures

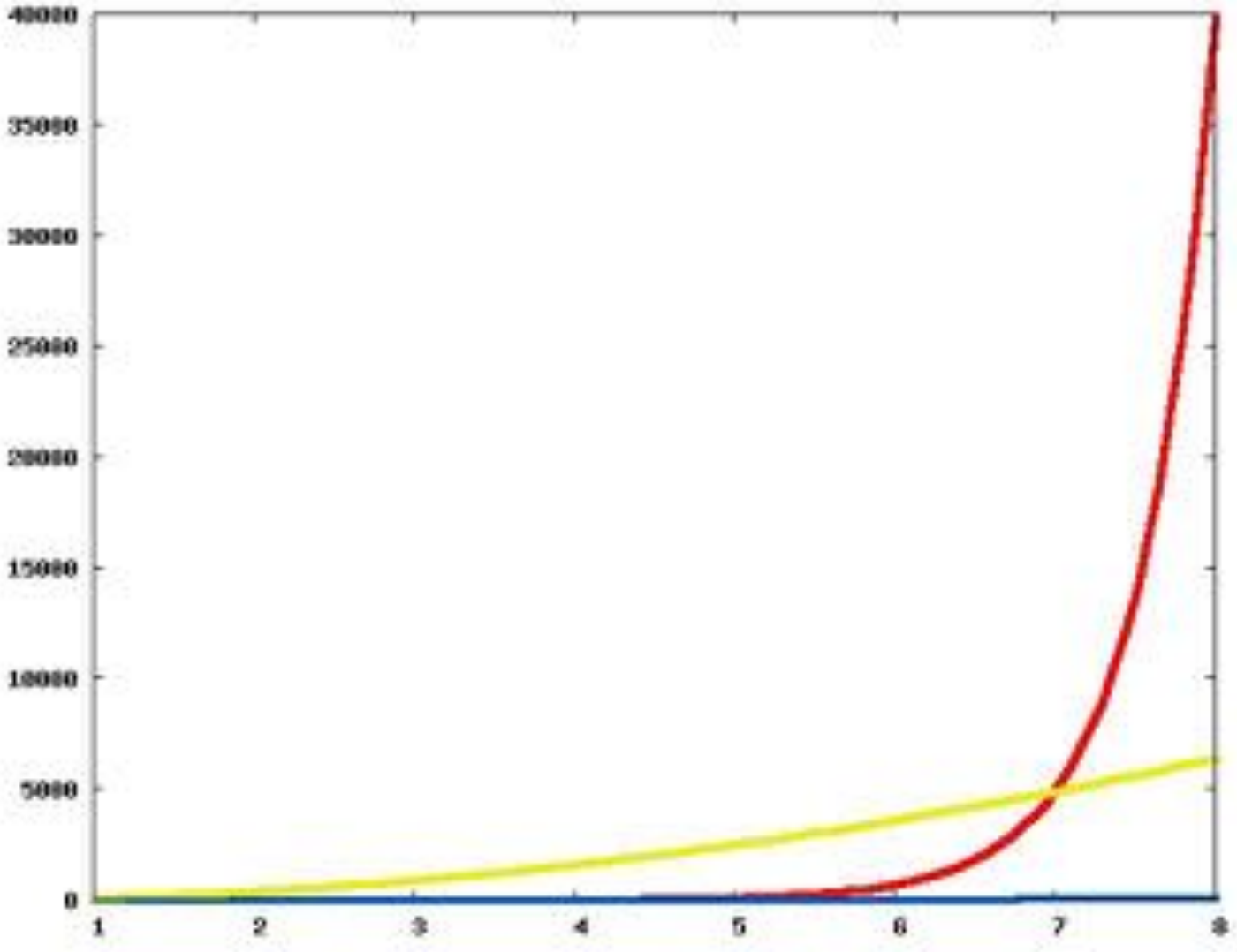
edit · good note | 0

Updated 3 days ago by Atri Rudra

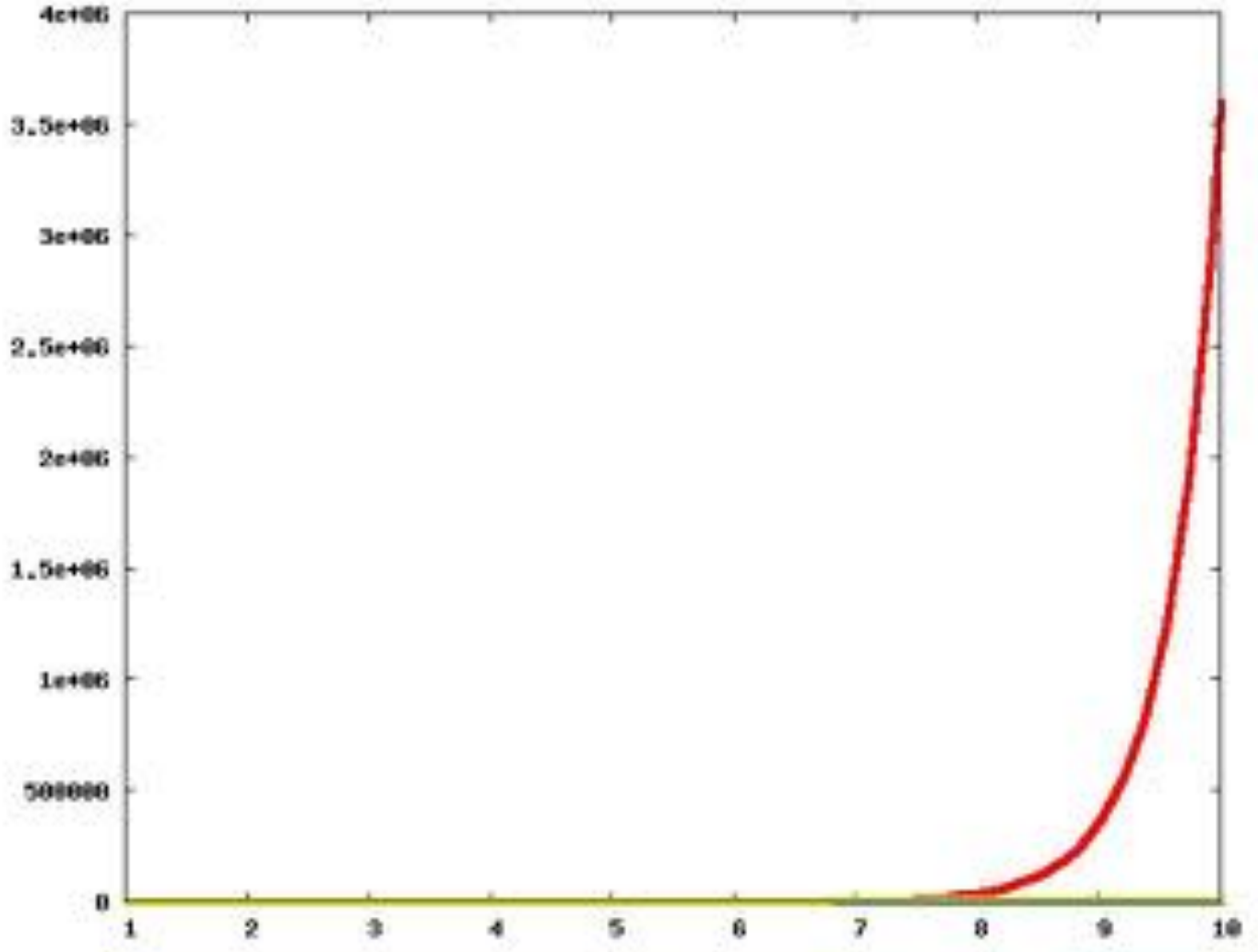
Which one is better?



Now?



And now?

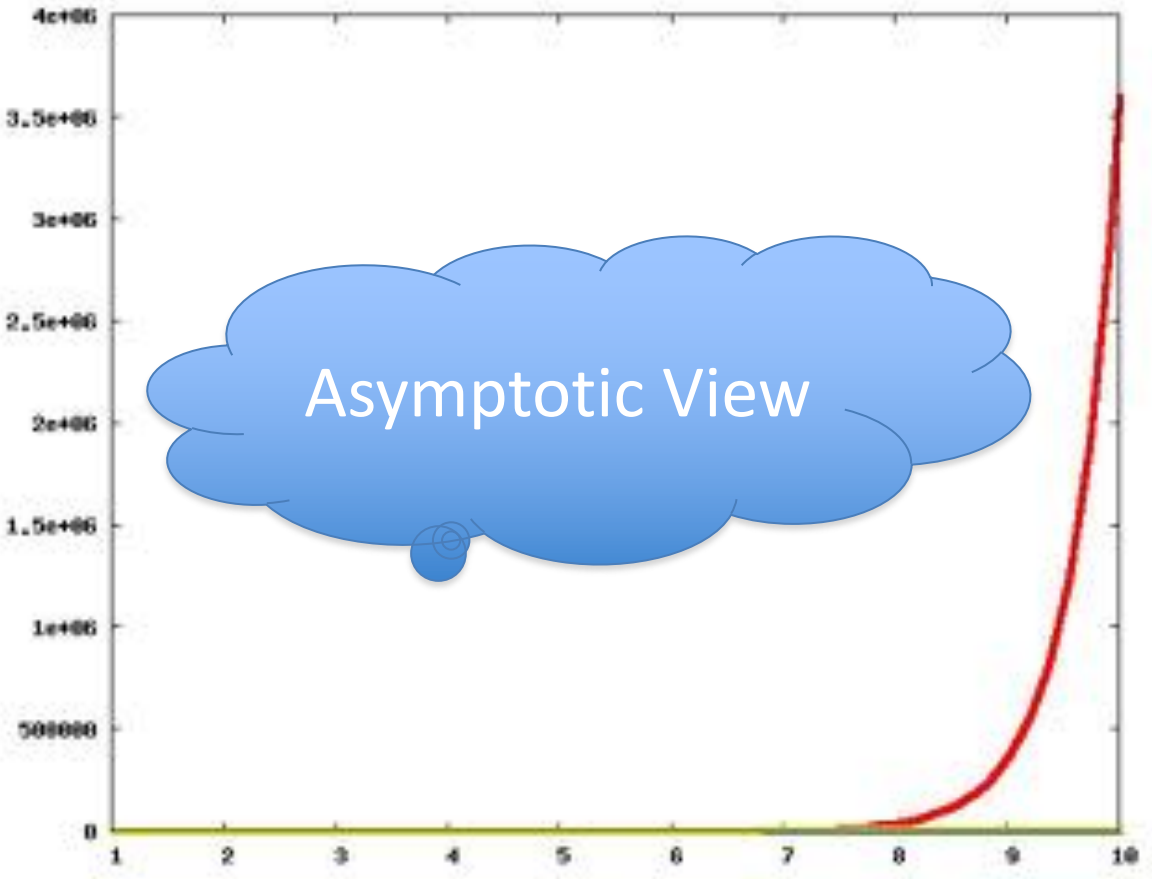


The actual run times

n!

$100n^2$

n^2



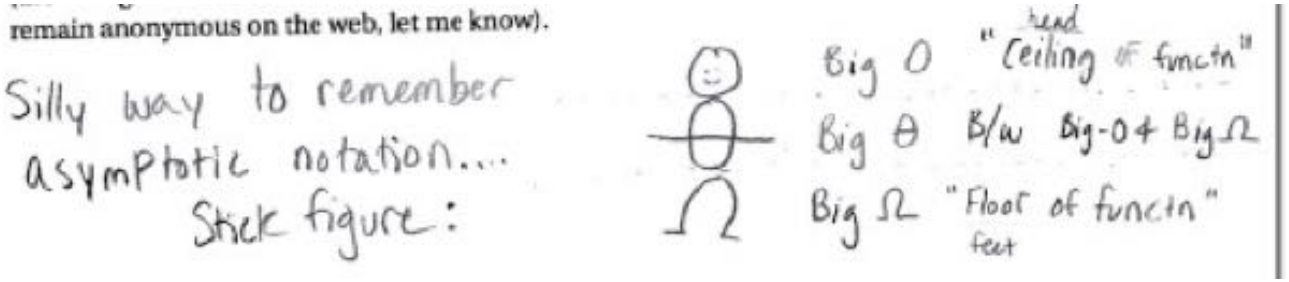
Asymptotic Notation

≤ is O with glasses

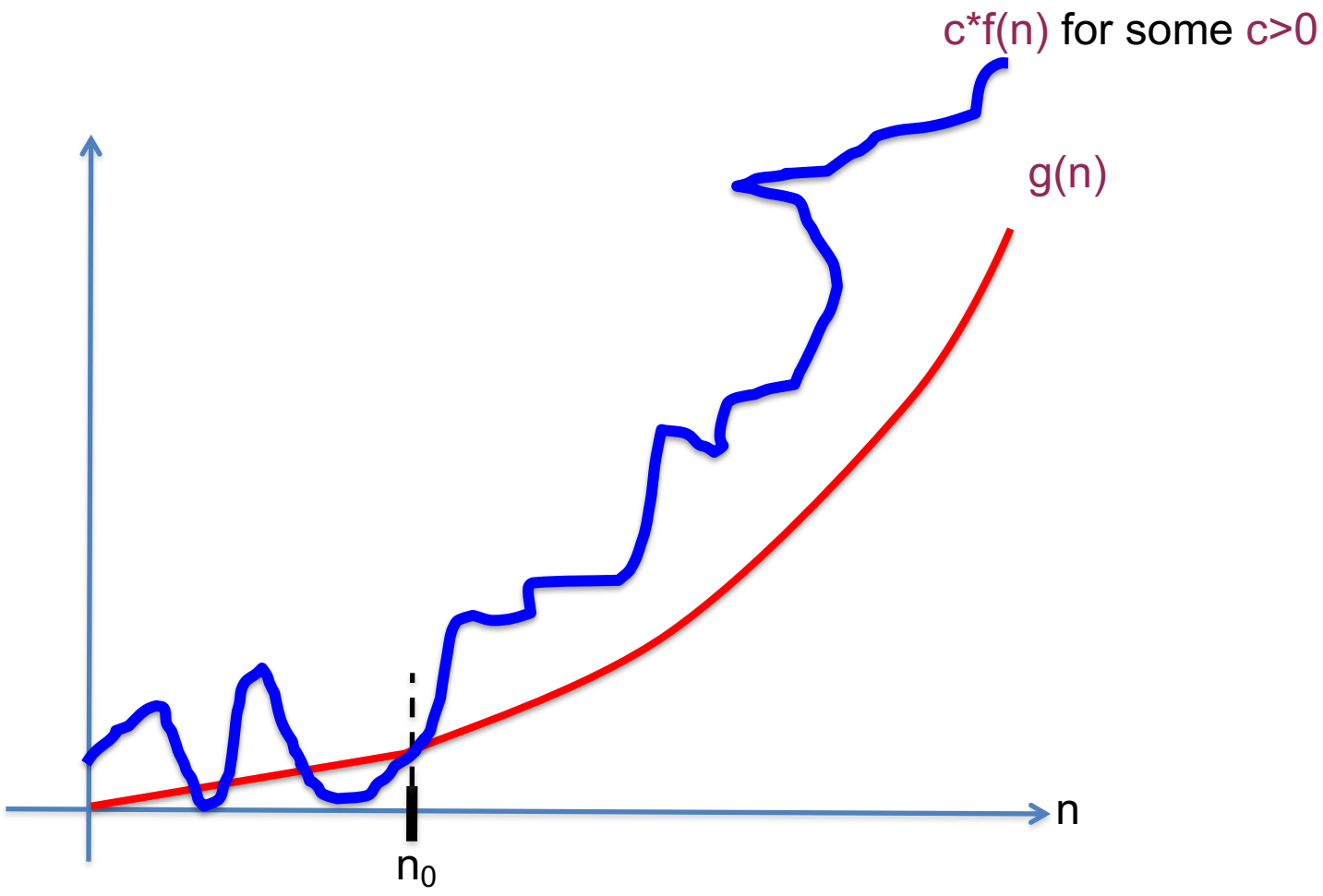
≥ is Ω with glasses

= is Θ with glasses

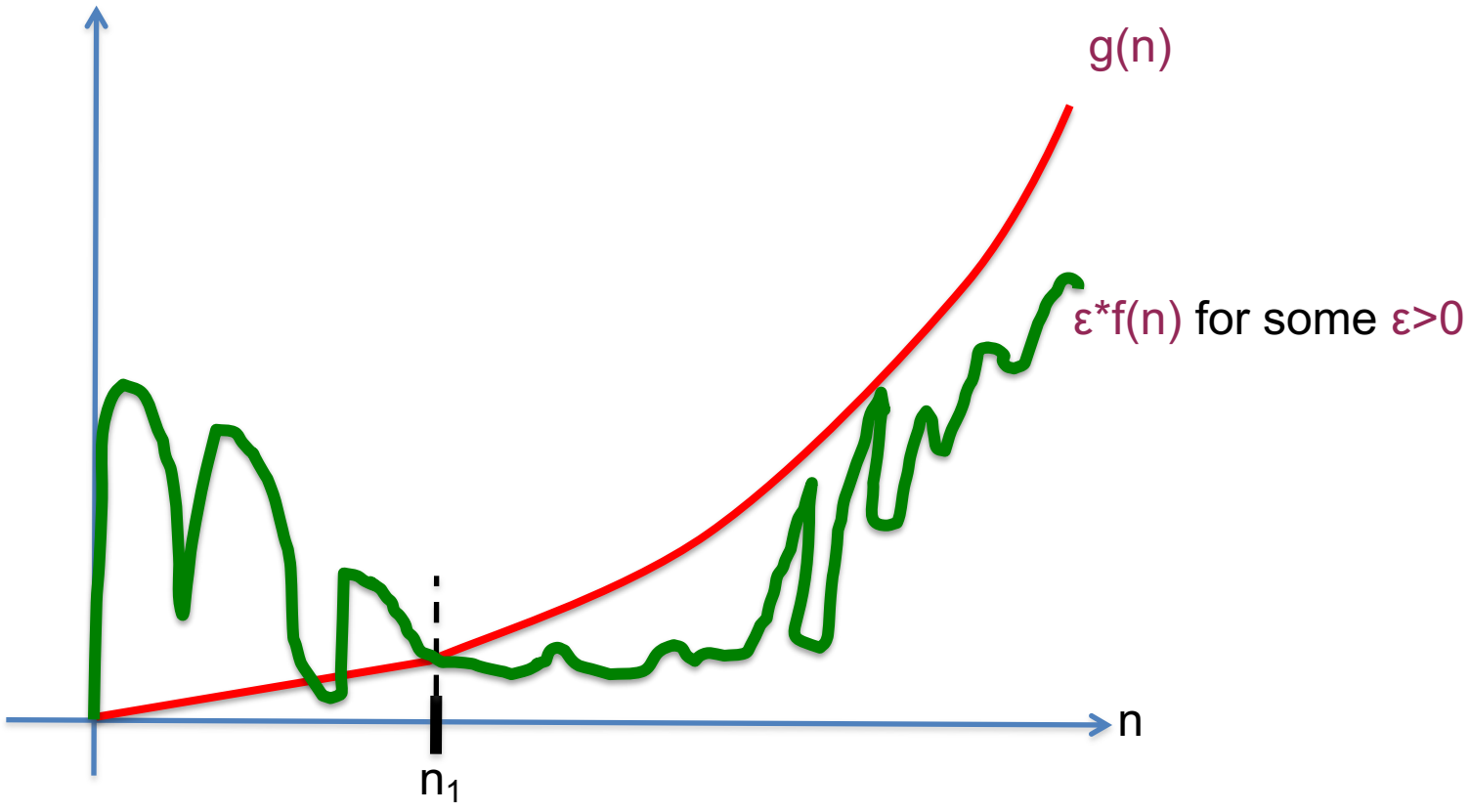
Another view



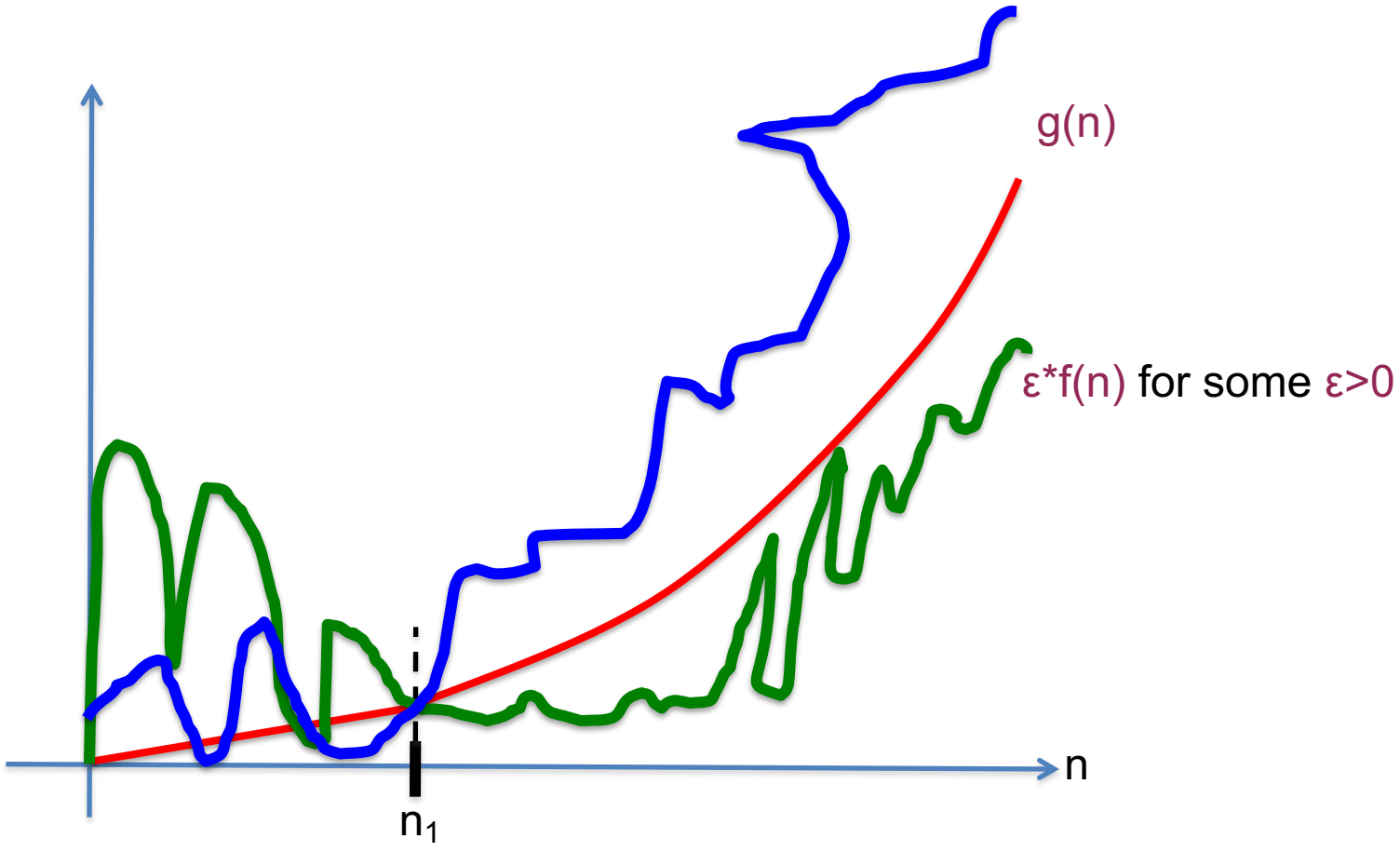
g(n) is O(f(n))



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