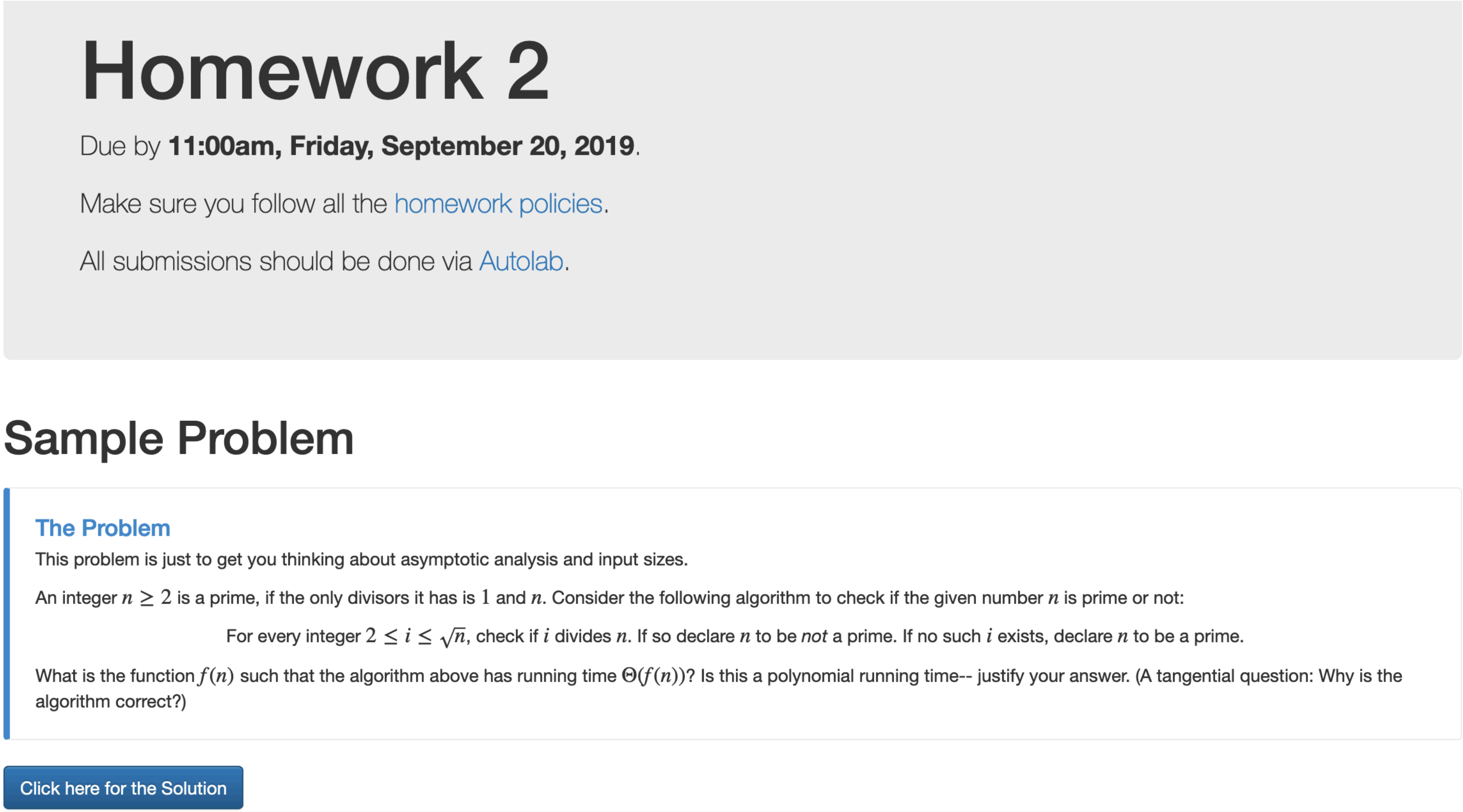
Lecture 8

CSE 331

Sep 13, 2018

HW 2 has been posted



Homework 2

Due by **11:00am, Friday, September 20, 2019**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

Sample Problem

The Problem

This problem is just to get you thinking about asymptotic analysis and input sizes.

An integer $n \geq 2$ is a prime, if the only divisors it has is 1 and n. Consider the following algorithm to check if the given number n is prime or not:

For every integer $2 \leq i \leq \sqrt{n}$, check if i divides n. If so declare n to be *not* a prime. If no such i exists, declare n to be a prime.

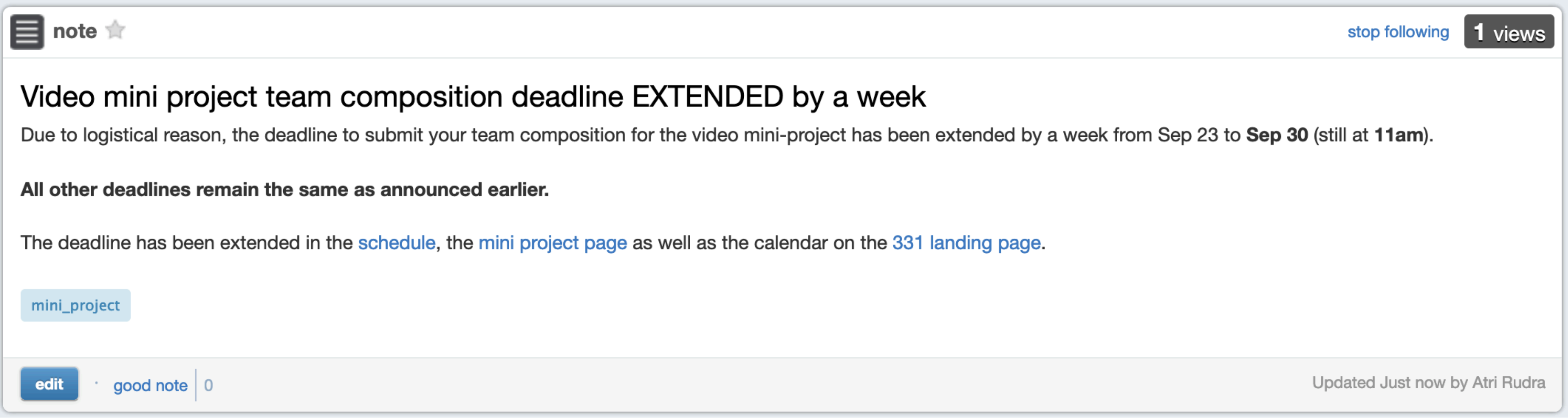
What is the function $f(n)$ such that the algorithm above has running time $\Theta(f(n))$? Is this a polynomial running time-- justify your answer. (A tangential question: Why is the algorithm correct?)

Solutions to HW1

Handed out at the end of the lecture

If you need it, ask for help

Mini project choice due on SEP 30



Video mini project team composition deadline EXTENDED by a week

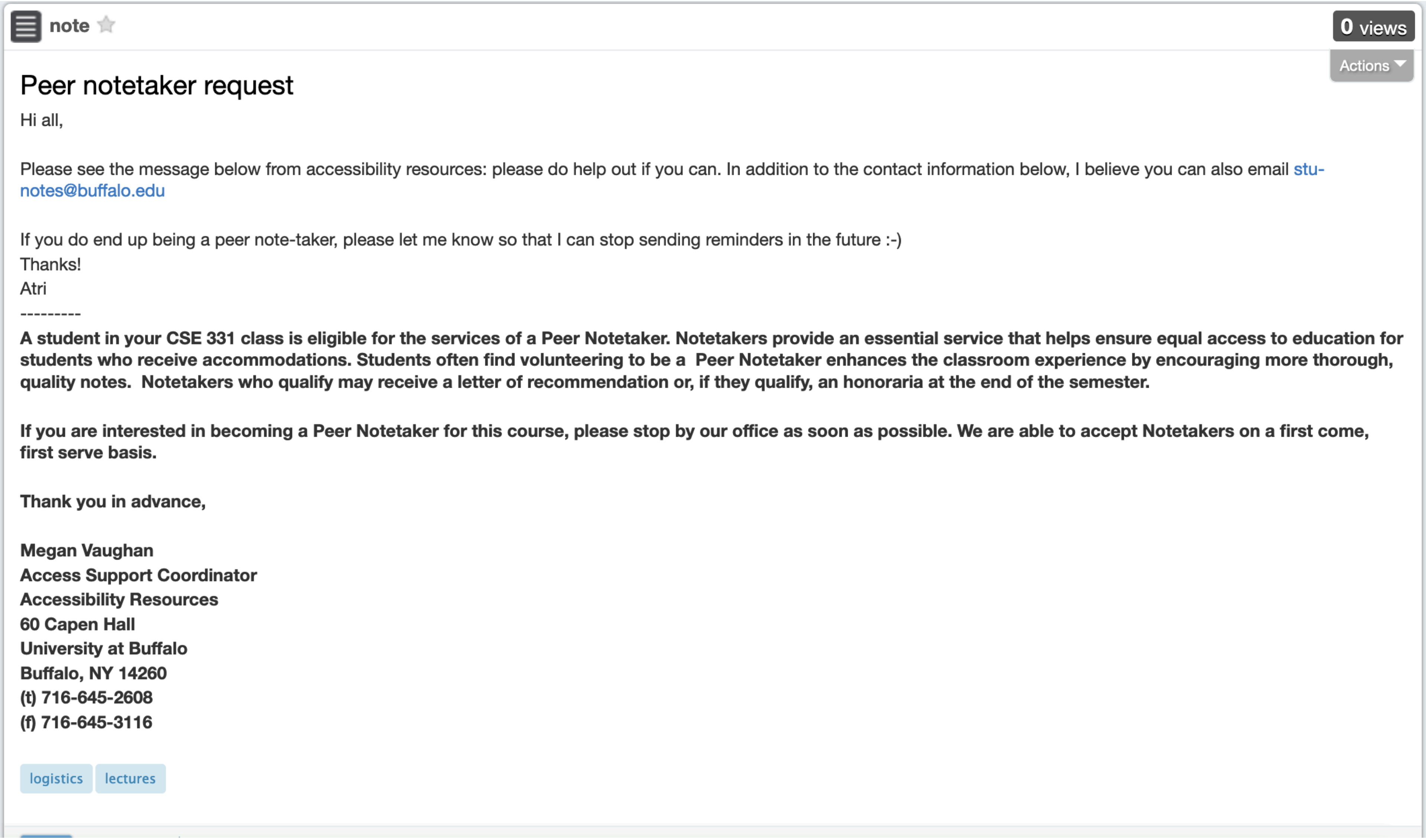
Due to logistical reason, the deadline to submit your team composition for the video mini-project has been extended by a week from Sep 23 to **Sep 30** (still at **11am**).

All other deadlines remain the same as announced earlier.

The deadline has been extended in the schedule, the mini project page as well as the calendar on the 331 landing page.

Updated Just now by Atri Rudra

Peer notetaker request



Peer notetaker request

Hi all,

Please see the message below from accessibility resources: please do help out if you can. In addition to the contact information below, I believe you can also email stu-notes@buffalo.edu

If you do end up being a peer note-taker, please let me know so that I can stop sending reminders in the future :-)
Thanks!
Atri

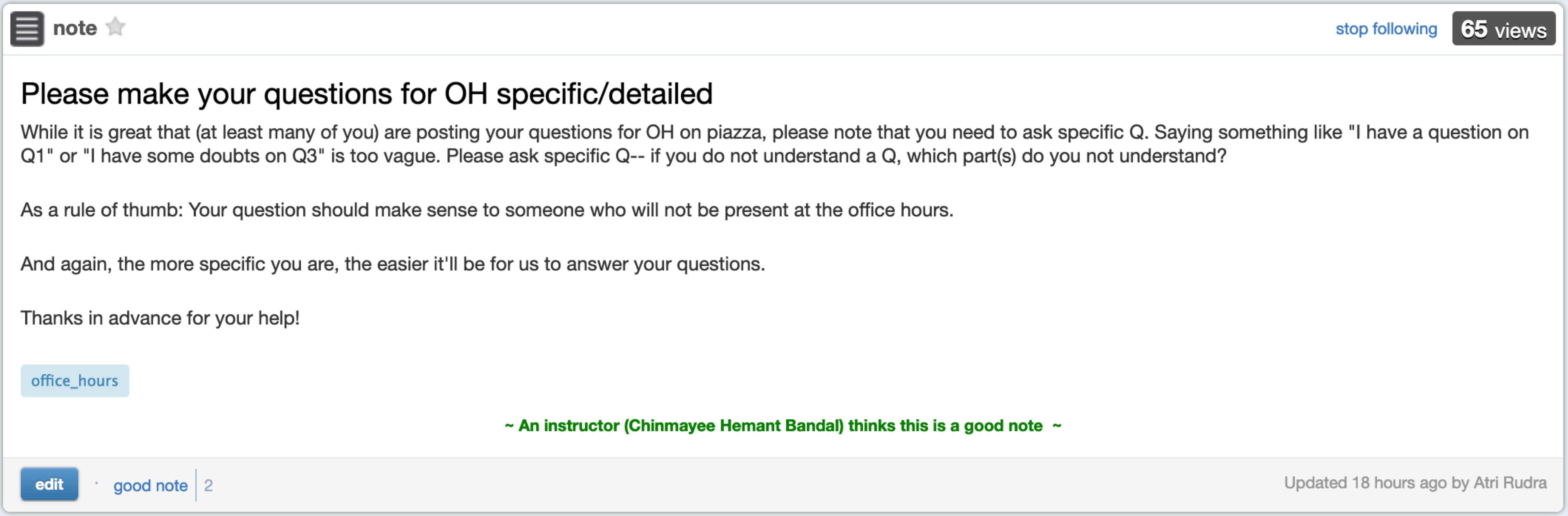
A student in your CSE 331 class is eligible for the services of a Peer Notetaker. Notetakers provide an essential service that helps ensure equal access to education for students who receive accommodations. Students often find volunteering to be a Peer Notetaker enhances the classroom experience by encouraging more thorough, quality notes. Notetakers who qualify may receive a letter of recommendation or, if they qualify, an honoraria at the end of the semester.

If you are interested in becoming a Peer Notetaker for this course, please stop by our office as soon as possible. We are able to accept Notetakers on a first come, first serve basis.

Thank you in advance,

Megan Vaughan
Access Support Coordinator
Accessibility Resources
60 Capen Hall
University at Buffalo
Buffalo, NY 14260
(t) 716-645-2608
(f) 716-645-3116

Updates to new OH policy



Please make your questions for OH specific/detailed

While it is great that (at least many of you) are posting your questions for OH on piazza, please note that you need to ask specific Q. Saying something like "I have a question on Q1" or "I have some doubts on Q3" is too vague. Please ask specific Q-- if you do not understand a Q, which part(s) do you not understand?

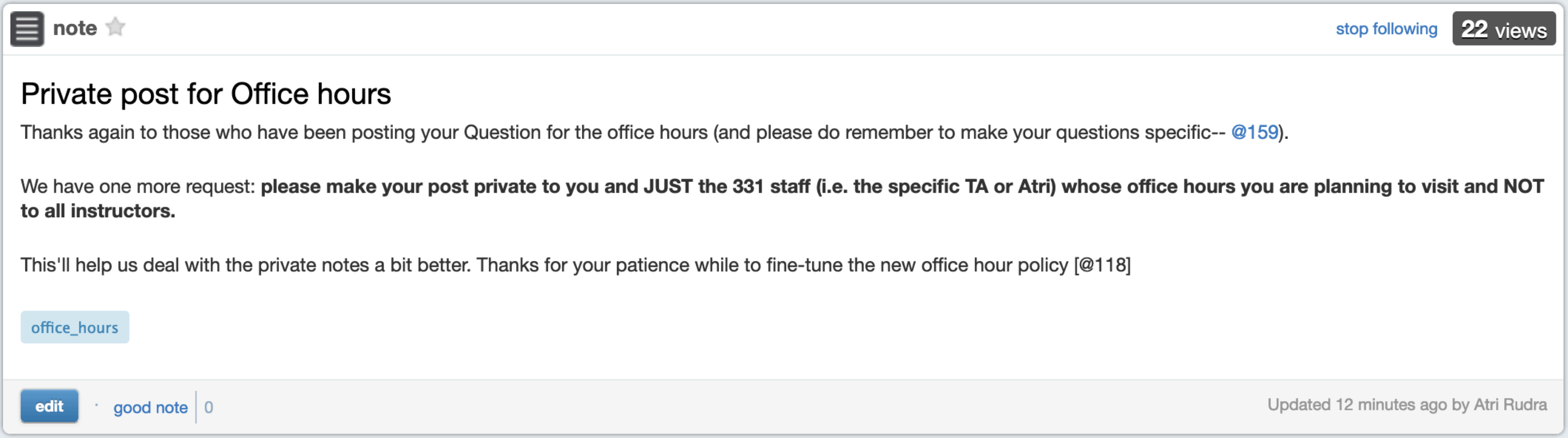
As a rule of thumb: Your question should make sense to someone who will not be present at the office hours.

And again, the more specific you are, the easier it'll be for us to answer your questions.

Thanks in advance for your help!

~ An instructor (Chinmayee Hemant Bandal) thinks this is a good note ~

Updated 18 hours ago by Atri Rudra



Private post for Office hours

Thanks again to those who have been posting your Question for the office hours (and please do remember to make your questions specific-- @159).

We have one more request: **please make your post private to you and JUST the 331 staff (i.e. the specific TA or Atri) whose office hours you are planning to visit and NOT to all instructors.**

This'll help us deal with the private notes a bit better. Thanks for your patience while to fine-tune the new office hour policy [@118]

Updated 12 minutes ago by Atri Rudra

Questions/Comments?

Observation 1

```
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
```

Once a man gets engaged, he remains engaged (to “better” women)

Observation 2

```
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 set S of engaged pairs as the final output
```

If w proposes to m after m', then she prefers m' to m

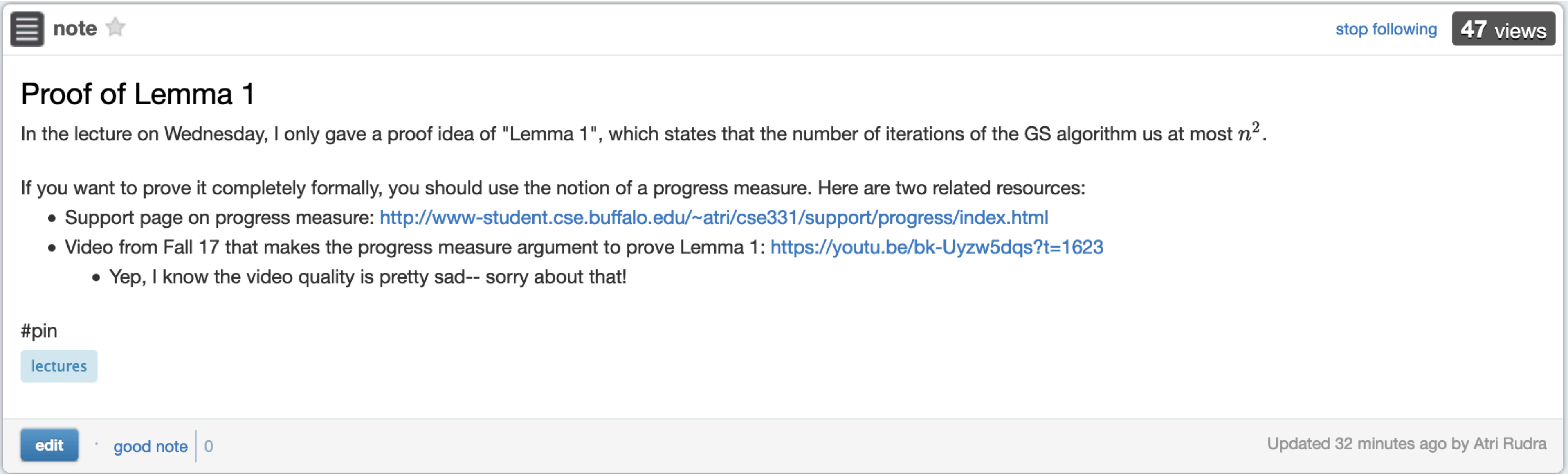
The Lemmas

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



Proof of Lemma 1

In the lecture on Wednesday, I only gave a proof idea of "Lemma 1", which states that the number of iterations of the GS algorithm us at most n^2.

If you want to prove it completely formally, you should use the notion of a progress measure. Here are two related resources:

- Support page on progress measure: http://www-student.cse.buffalo.edu/~atri/cse331/support/progress/index.html
- Video from Fall 17 that makes the progress measure argument to prove Lemma 1: https://youtu.be/bk-Uyzw5dqs?t=1623
  - Yep, I know the video quality is pretty sad-- sorry about that!

#pin

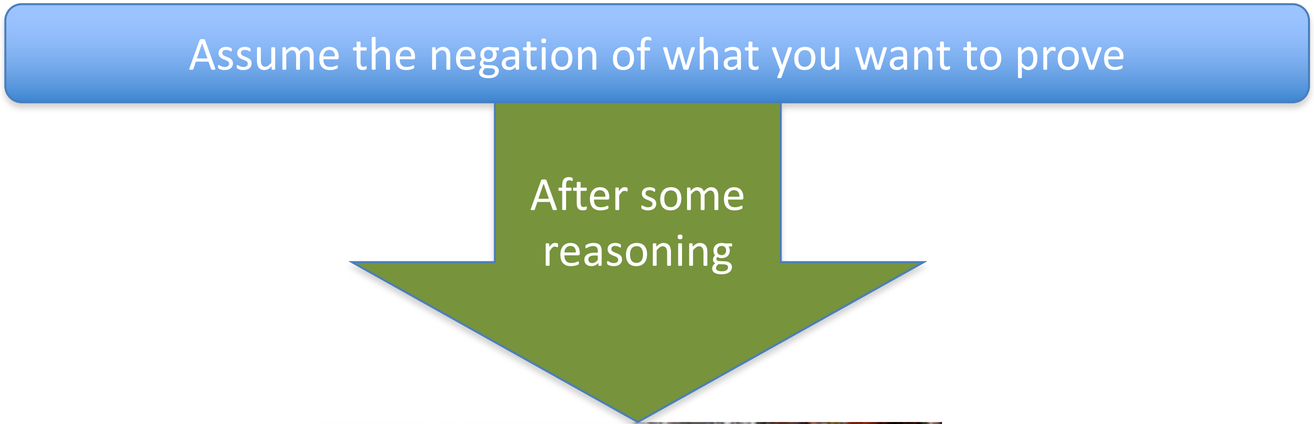
lectures

Updated 32 minutes ago by Atri Rudra

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 2

Lemma 4:

If at the end of an iteration, w is free

then w has **not** proposed to all men

Proof of Lemma 3

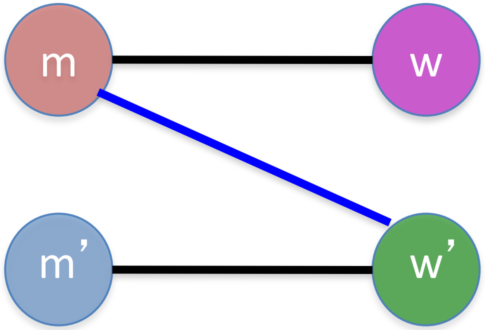
By contradiction

Assume there is an instability (m,w’)

w’ last proposed to m’

m prefers w’ to w

w’ prefers m to m’



Contradiction by Case Analysis

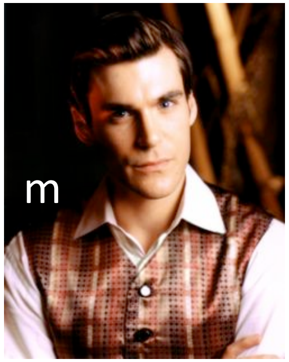
Depending on whether w’ had proposed to m or not

Case 1: w’ never proposed to m

w’ prefers m’ to m

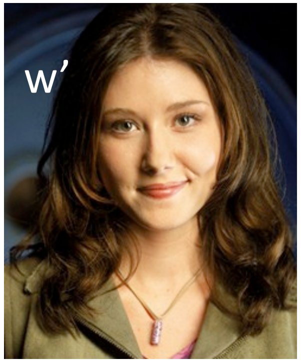
By Obs 2

Assumed w’ prefers m to m’



Source: 4simpsons.wordpress.com

Case 2: w' had proposed to m



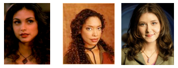
Case 2.1: m had accepted w' proposal

m is finally engaged to w

Thus, m prefers w to w'

By Obs 1

4simpsons.wordpress.com



Case 2.2: m had rejected w' proposal

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

By Obs 1

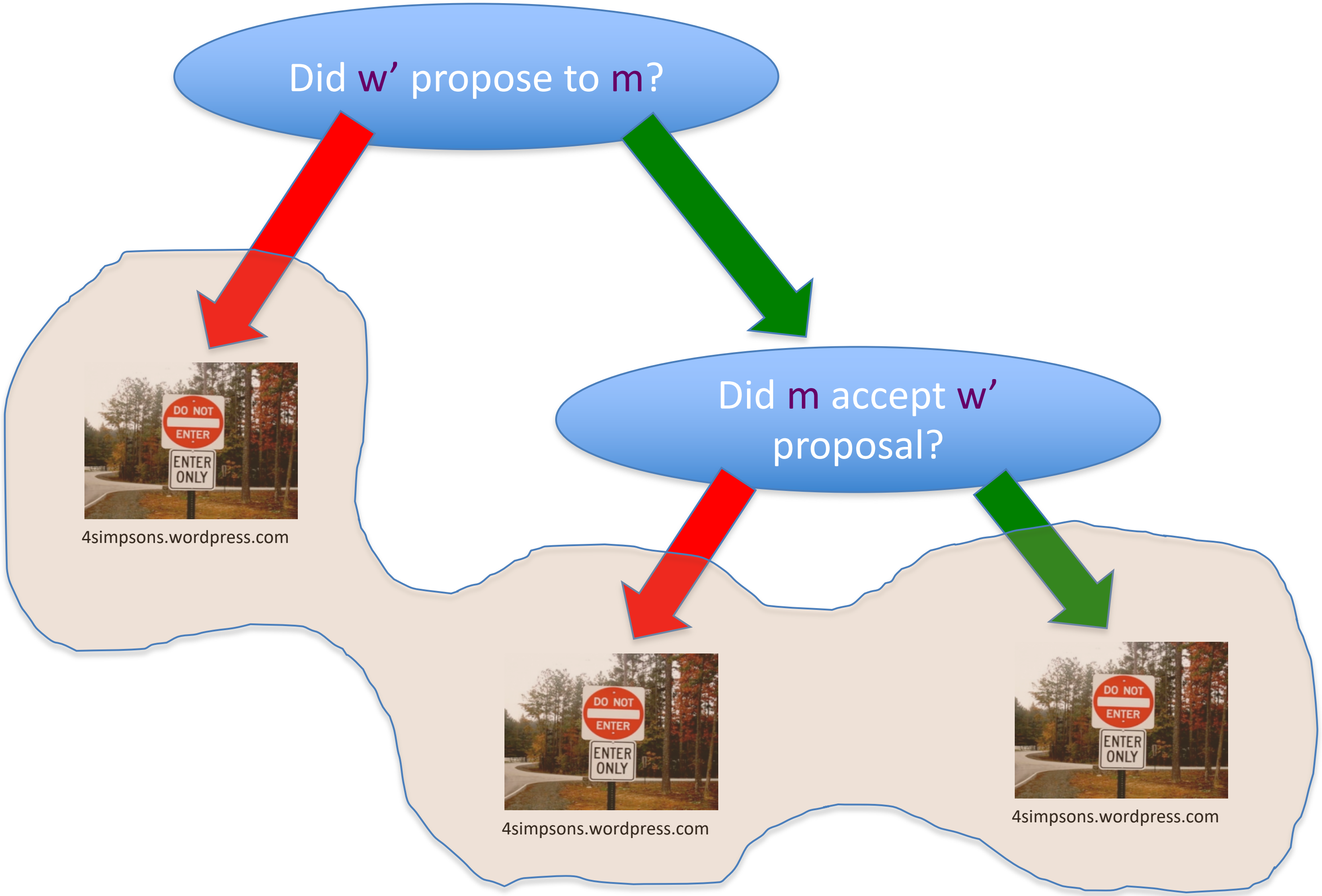
m is finally engaged to w (prefers w to w''

By Obs 1

m prefers w to w'

4simpsons.wordpress.com

Overall structure of case analysis



Did w’ propose to m?

Did m accept w’ proposal?

4simpsons.wordpress.com

4simpsons.wordpress.com

4simpsons.wordpress.com

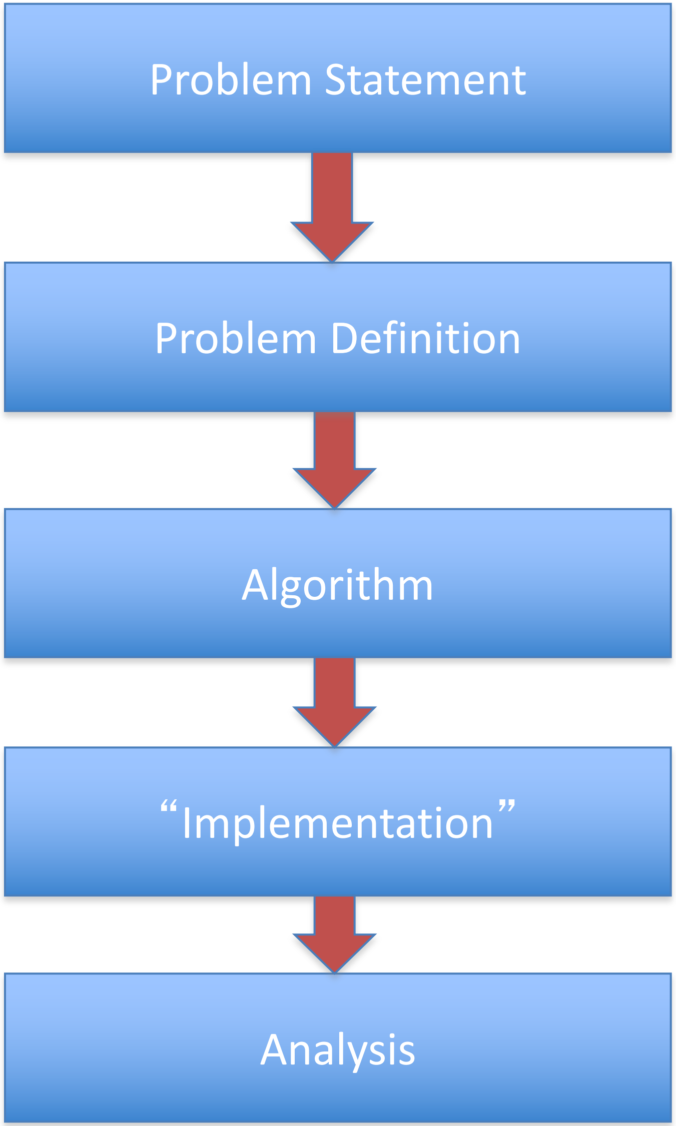
Questions?

Extensions

Fairness of the GS algorithm

Different executions of the GS algorithm

Main Steps in Algorithm Design



Problem Statement

Problem Definition

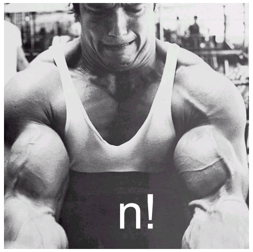
Algorithm

"Implementation"

Analysis

NRMP

National Resident Matching Program



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?

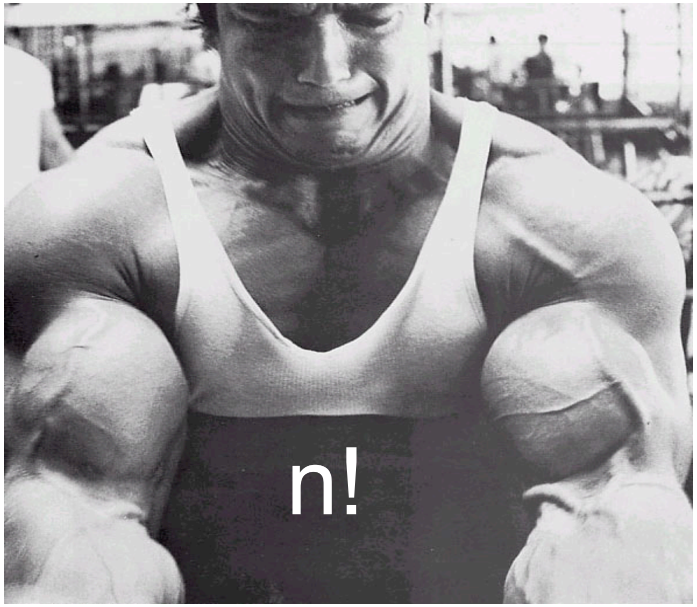
Worst-case Inputs

$N = 2n^2$ for SMP

Efficient in terms of what?

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)

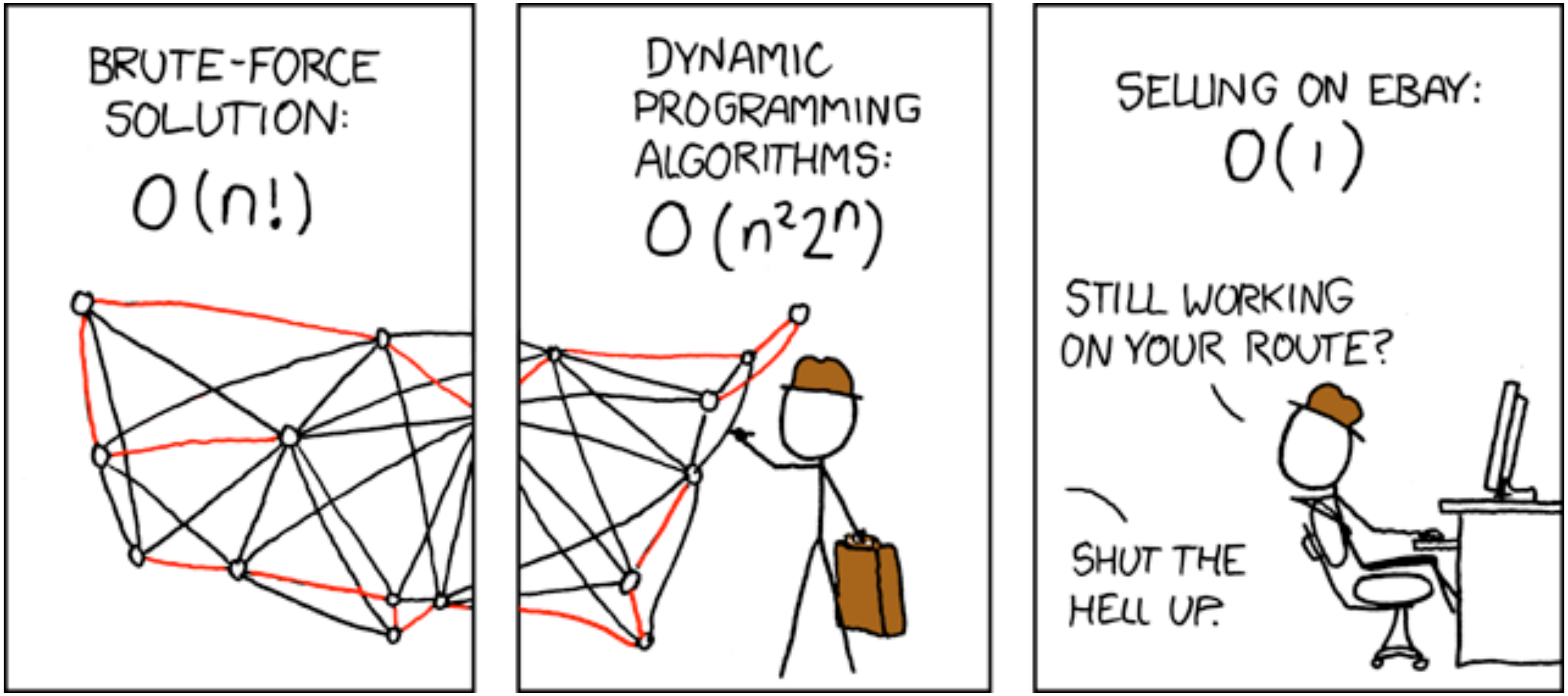
Step: “primitive computational step”

More on polynomial time

Problem centric tractability

Can talk about problems that are not efficient!

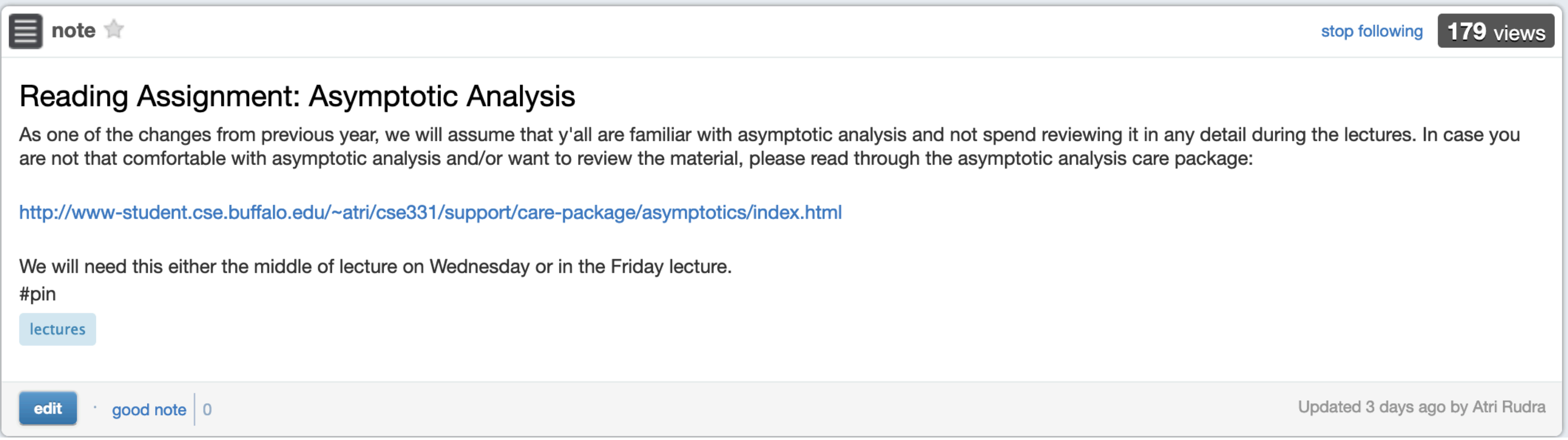
Asymptotic Analysis



Travelling Salesman Problem

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

Reading Assignment for today



Reading Assignment: Asymptotic Analysis

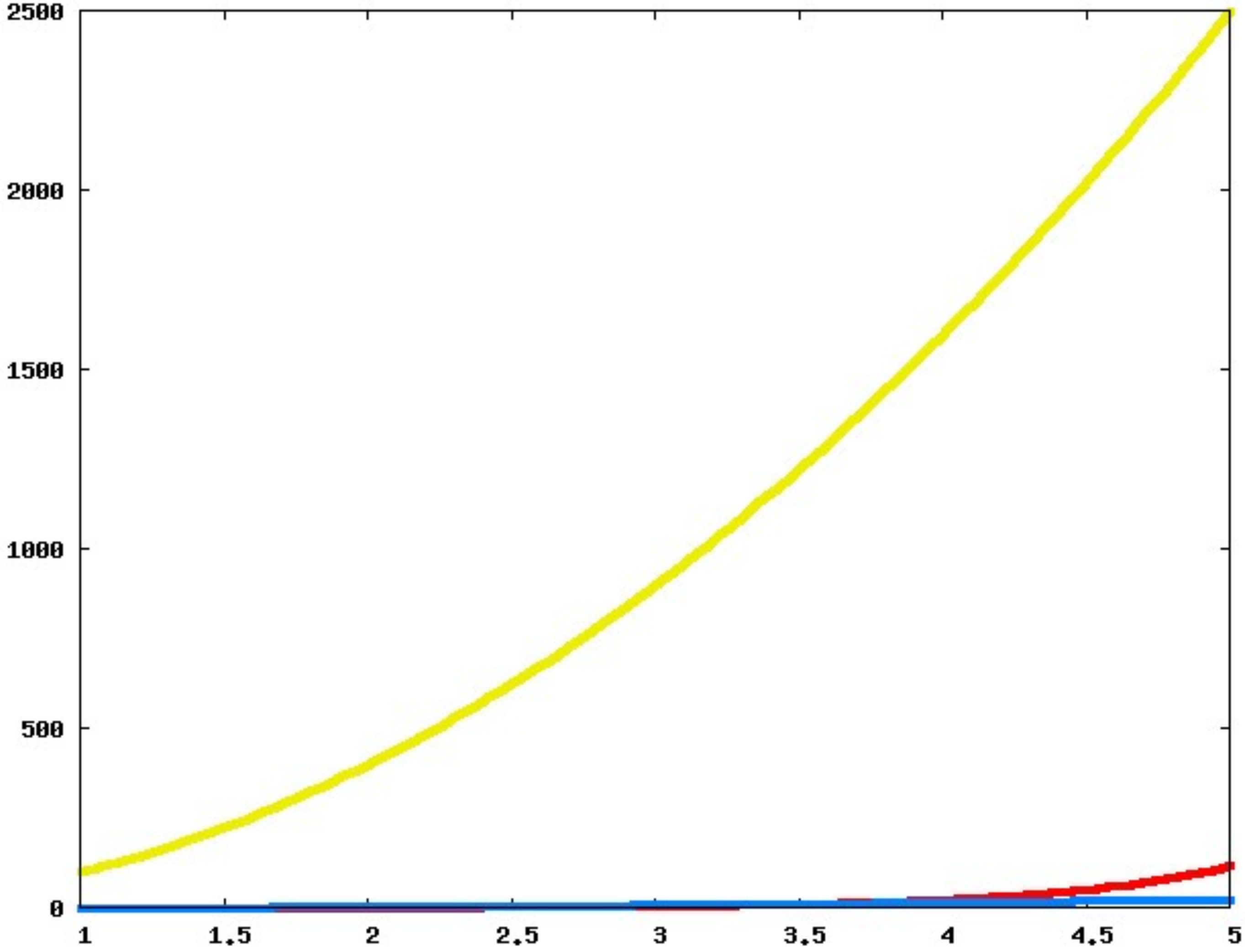
As one of the changes from previous year, we will assume that y'all are familiar with asymptotic analysis and not spend reviewing it in any detail during the lectures. In case you are not that comfortable with asymptotic analysis and/or want to review the material, please read through the asymptotic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

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

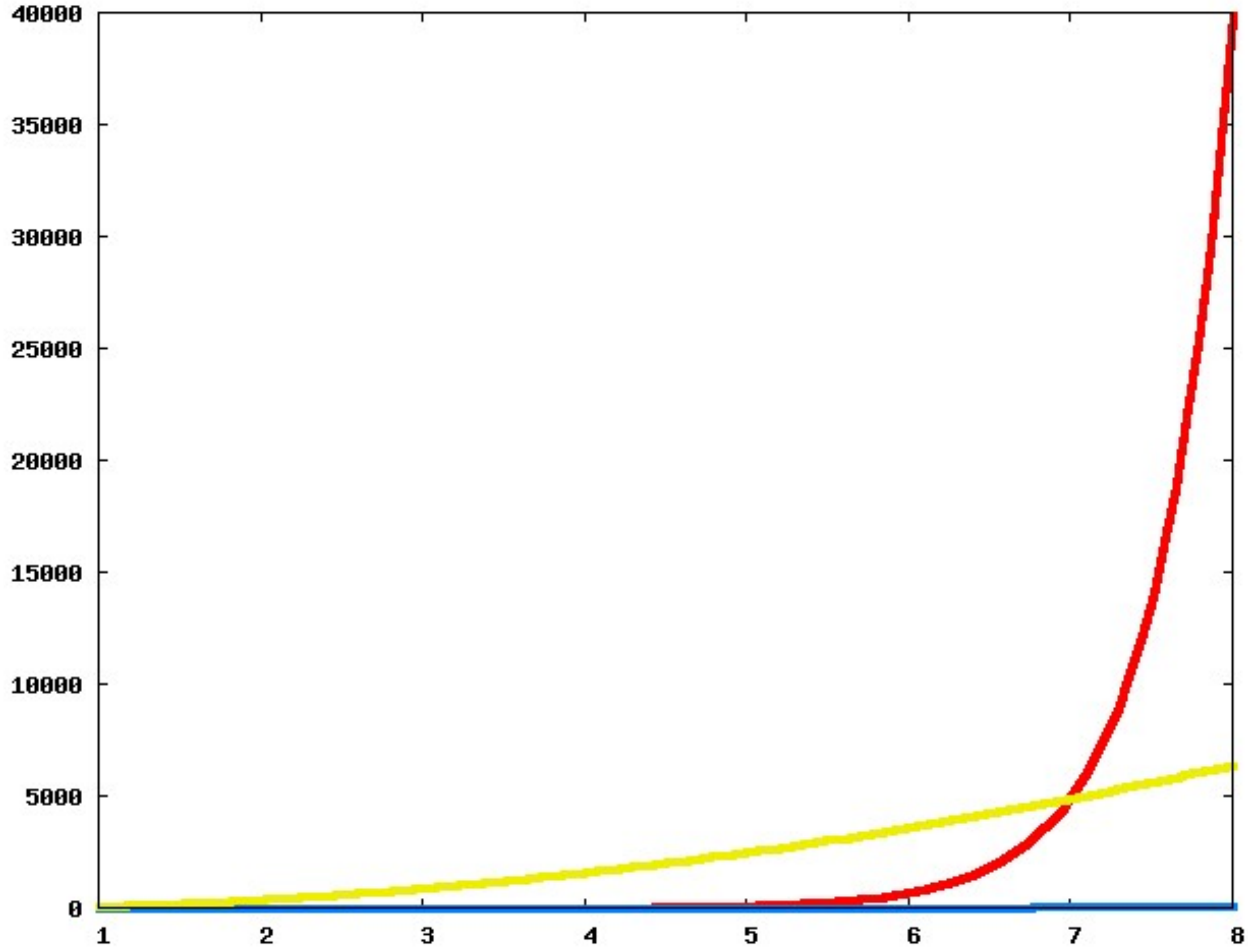
We will need this either the middle of lecture on Wednesday or in the Friday lecture.
#pin

Updated 3 days ago by Atri Rudra

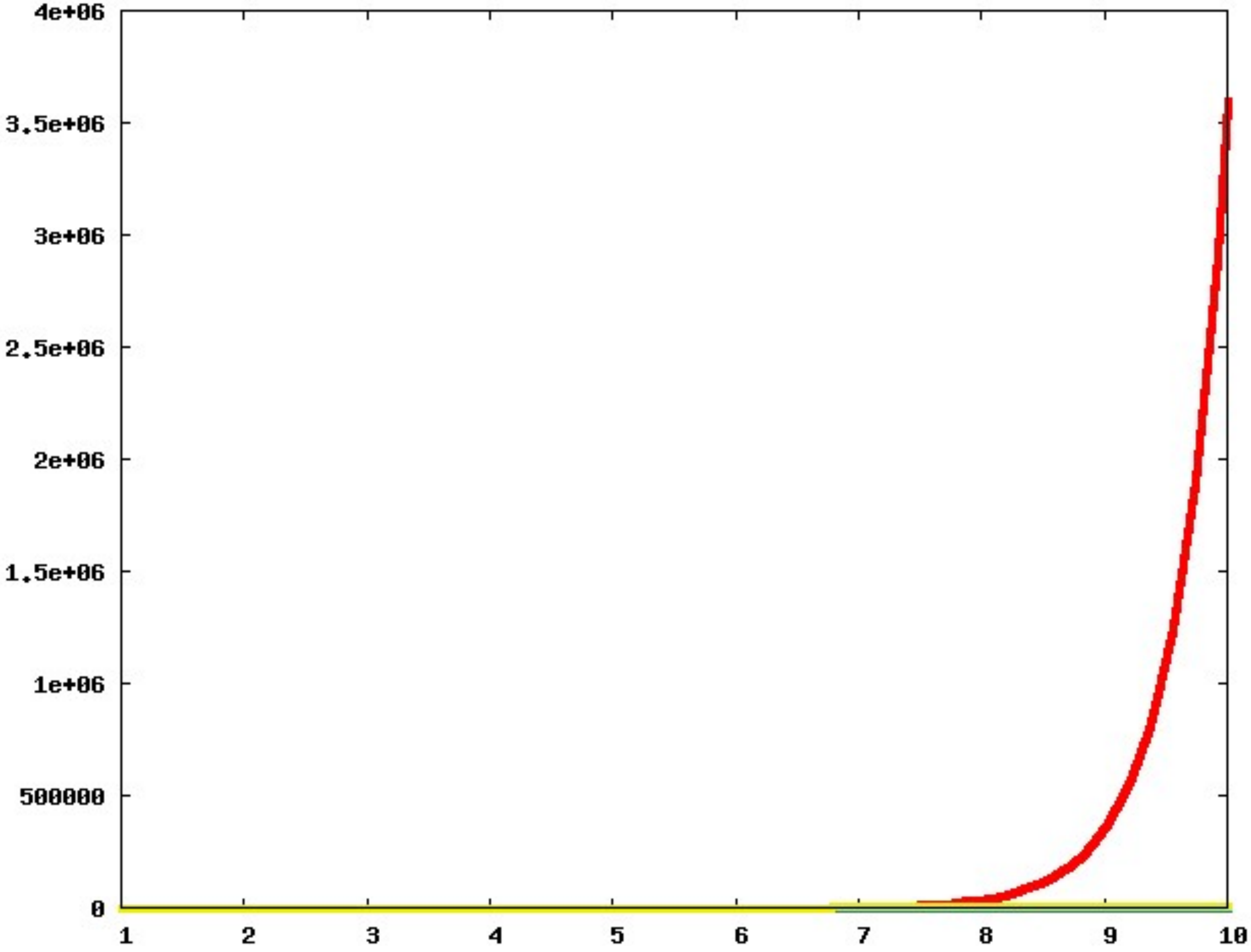
Which one is better?



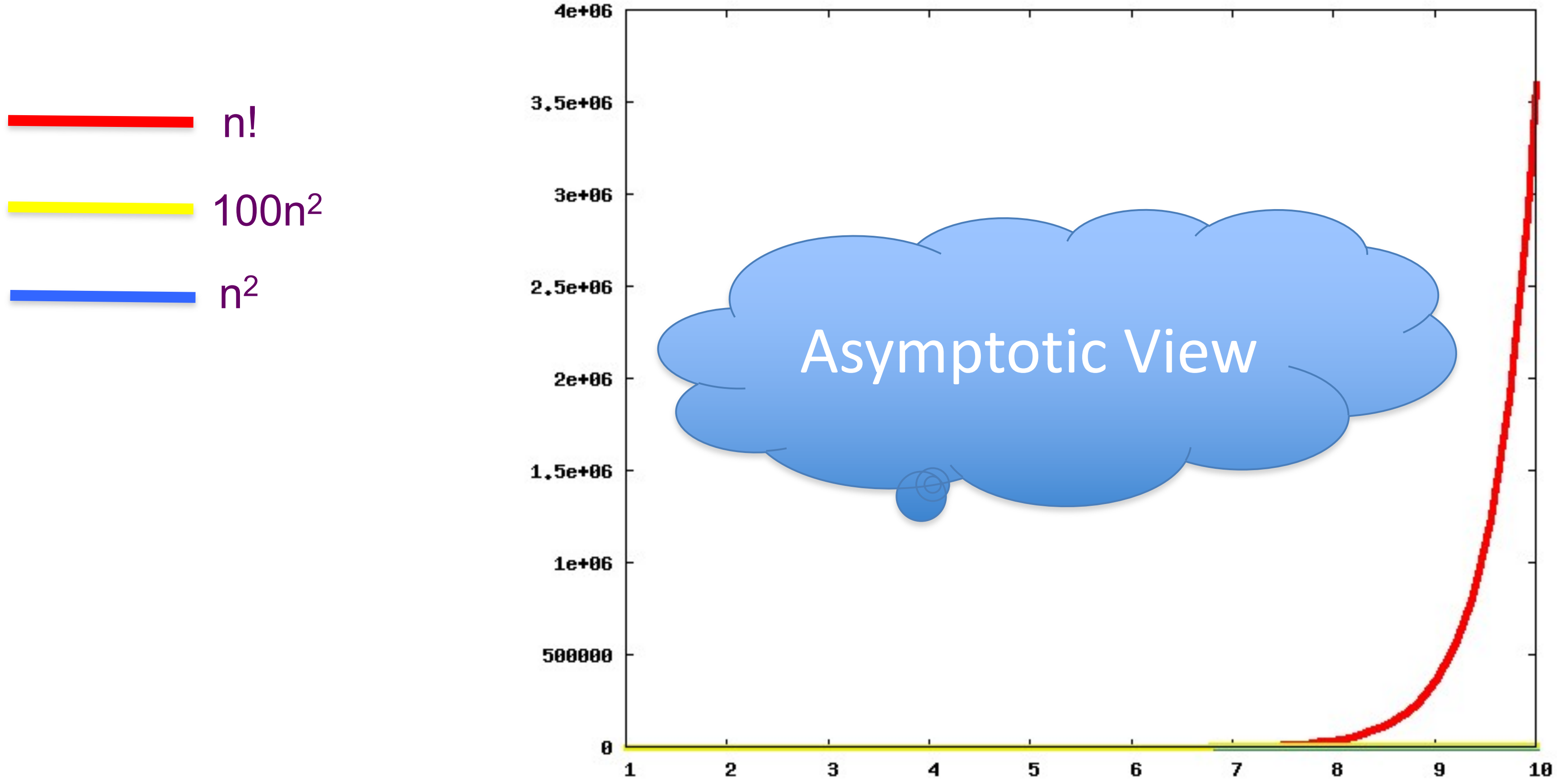
Now?



And now?

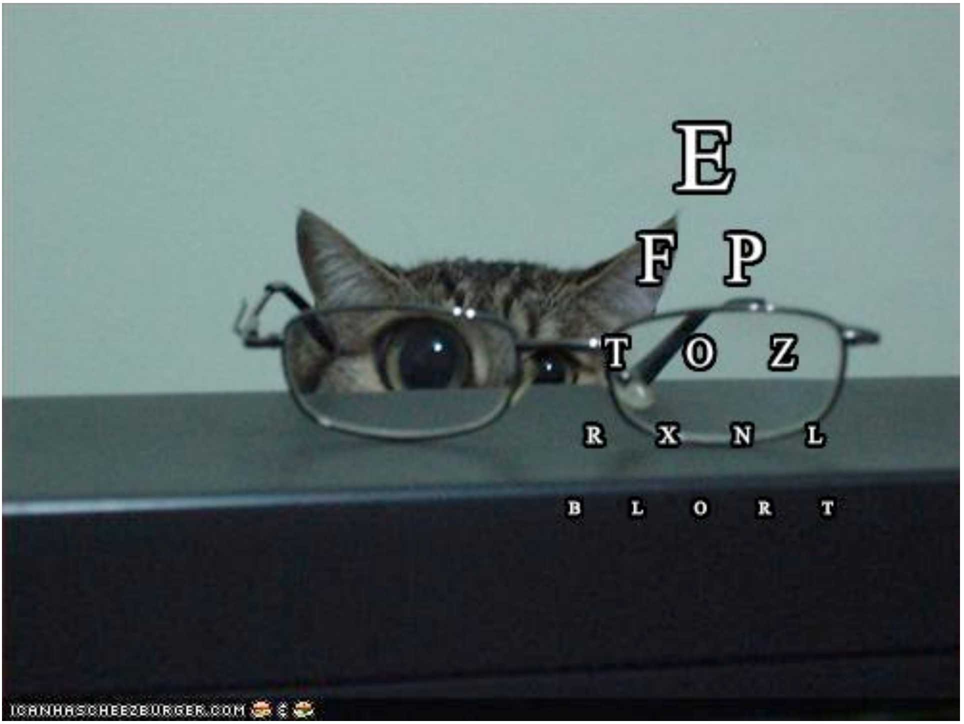


The actual run times



Asymptotic View

Asymptotic Notation

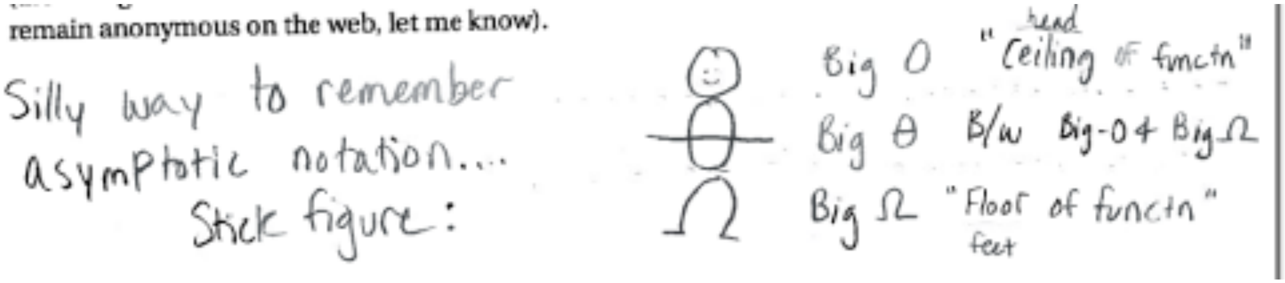


≤ is O with glasses

≥ is Ω with glasses

= is Θ with glasses

Another view



remain anonymous on the web, let me know).

Silly way to remember asymptotic notation.... Stick figure:

head

Big O "Ceiling of functn"

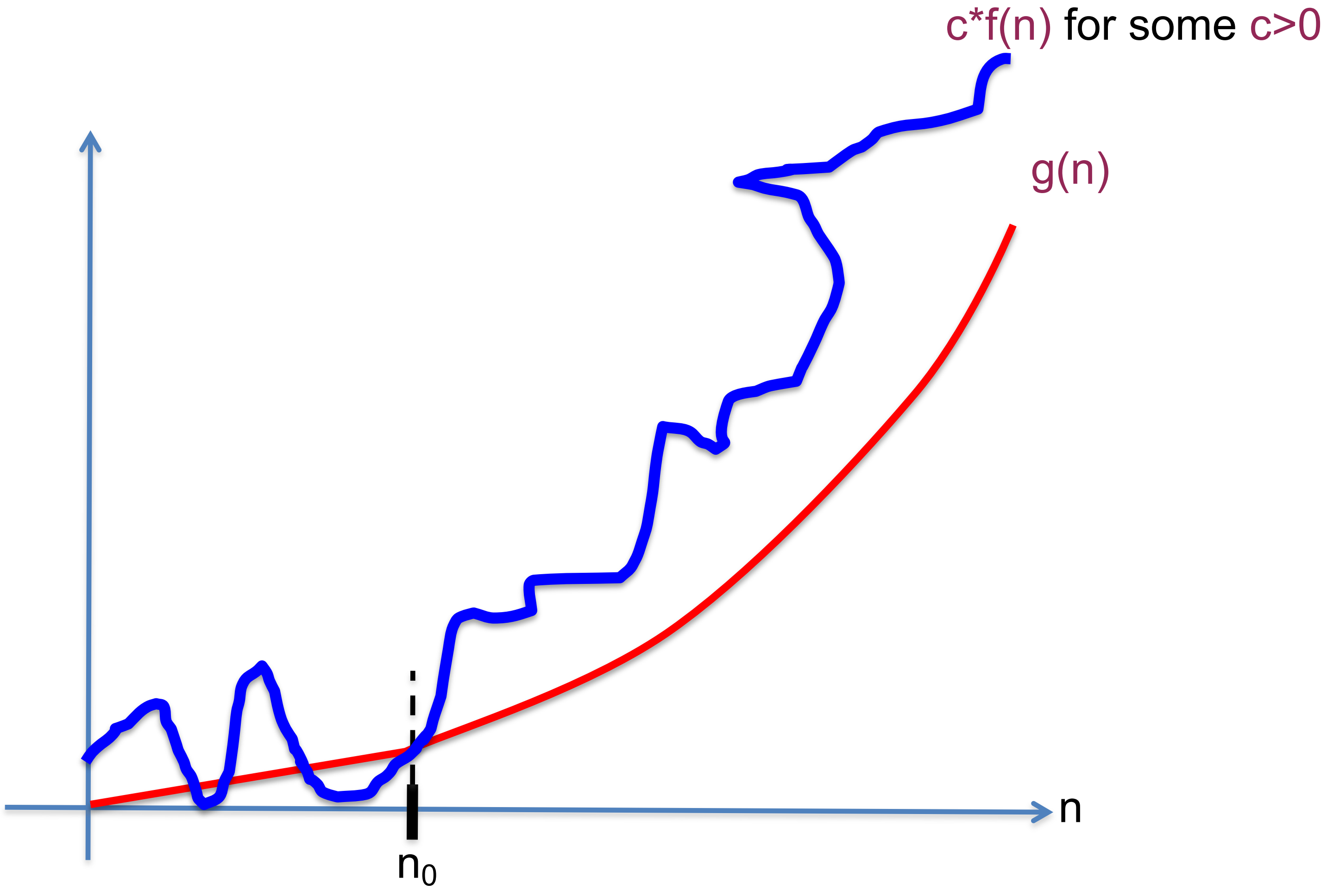
Big Θ B/w Big-O + Big Ω

Big Ω "Floor of functn"

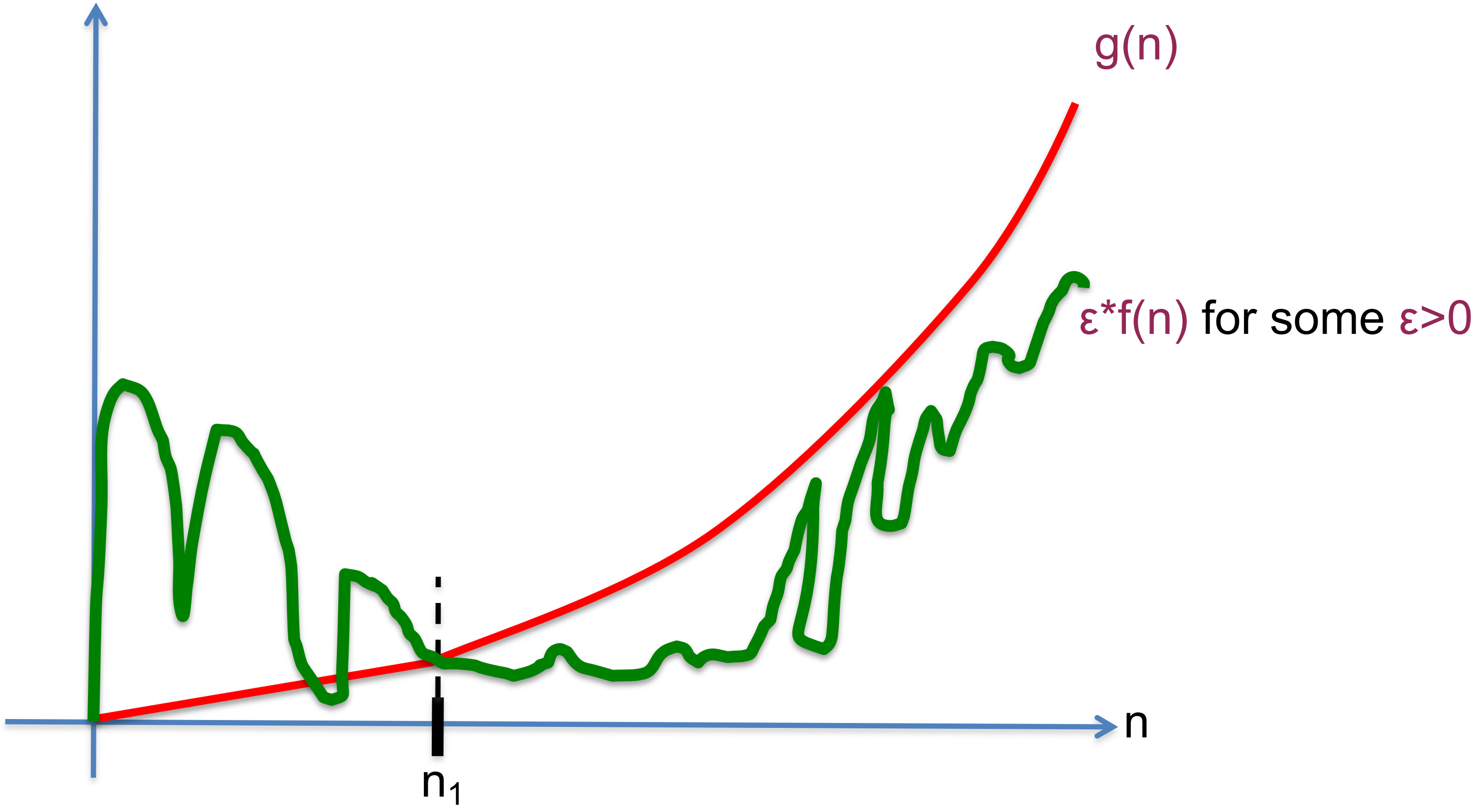
feet

© Aleksandra Patrzalek, 2012

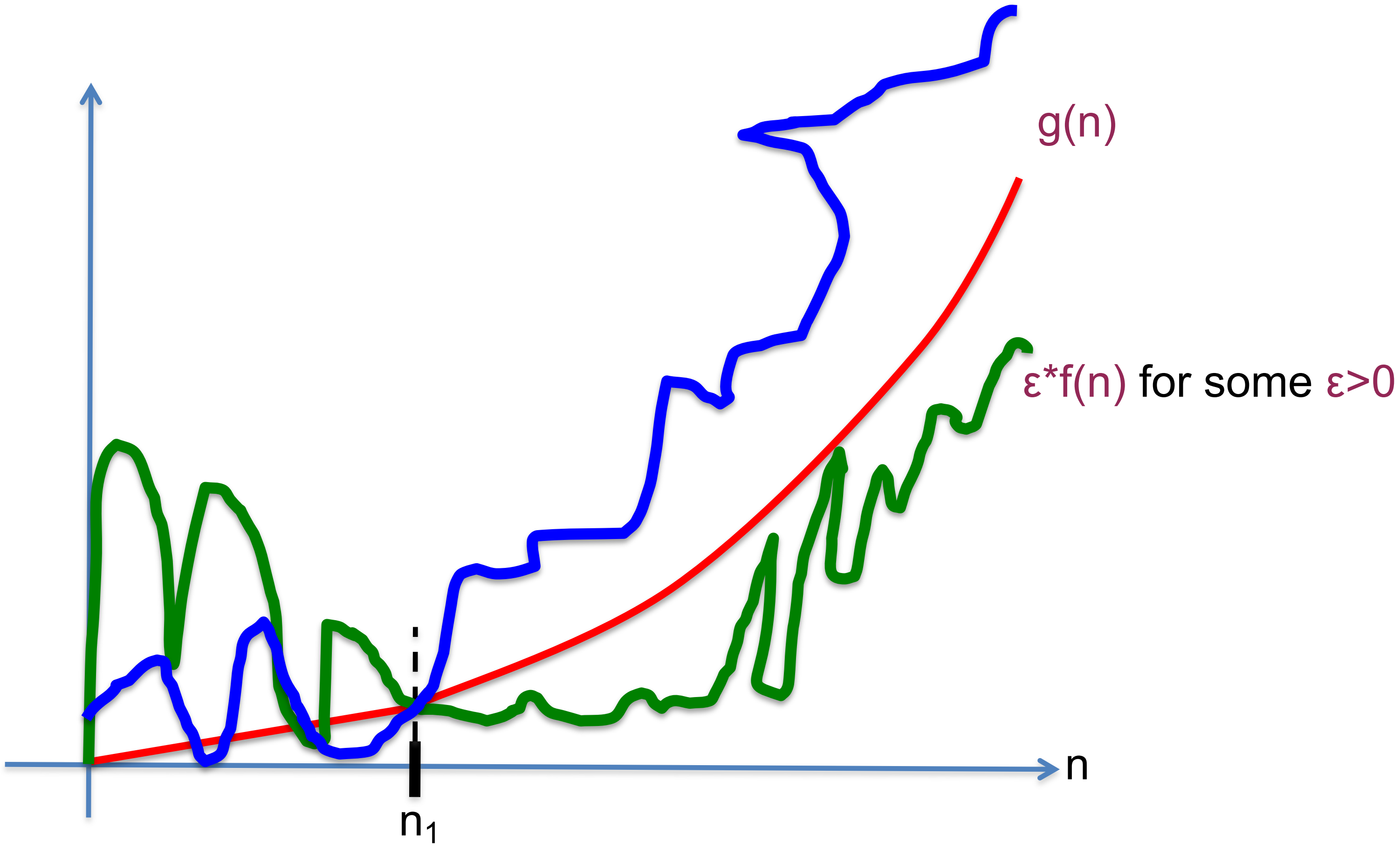
g(n) is O(f(n))



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

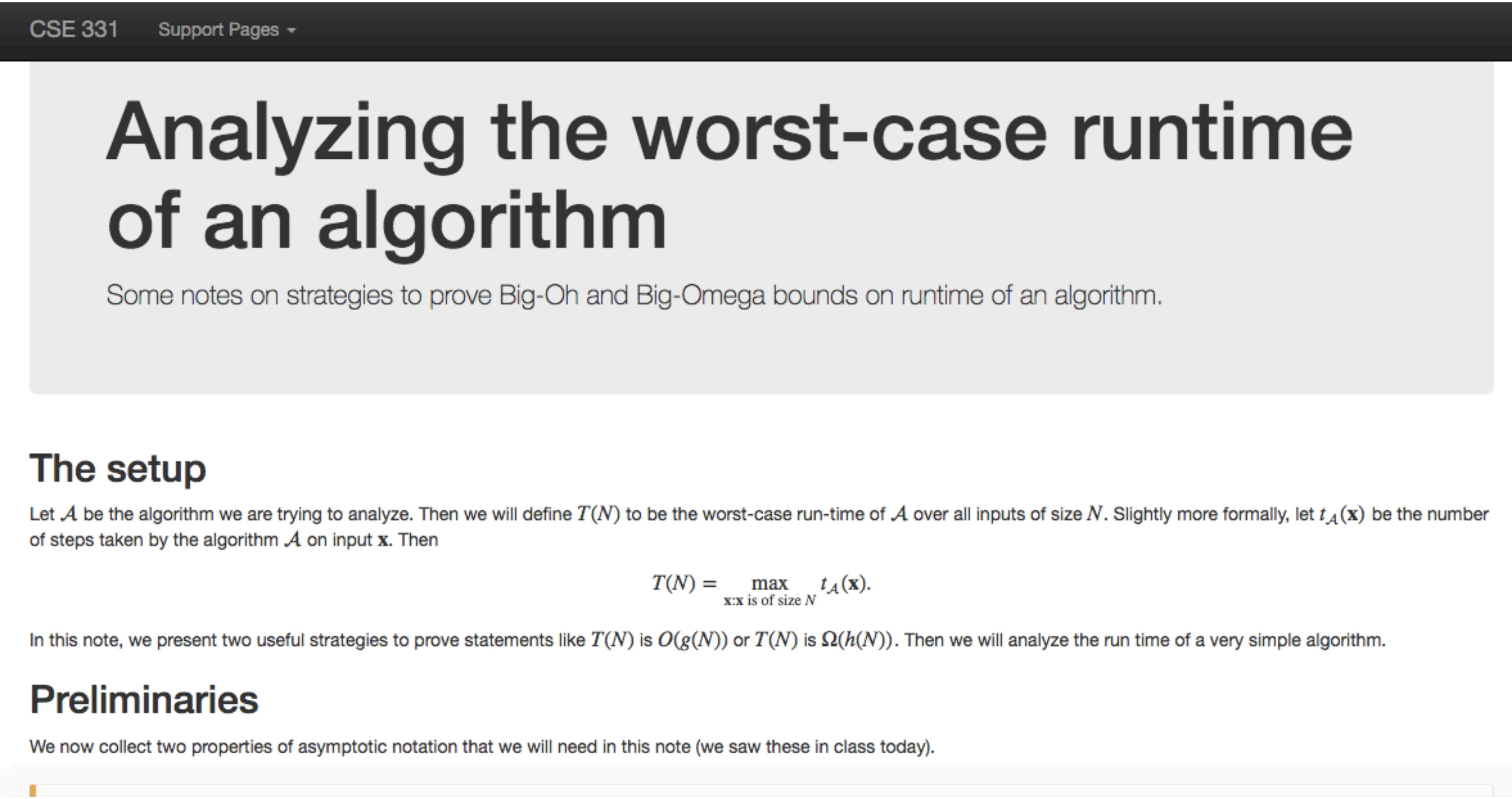
Multiplicative

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

Another Reading Assignment



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

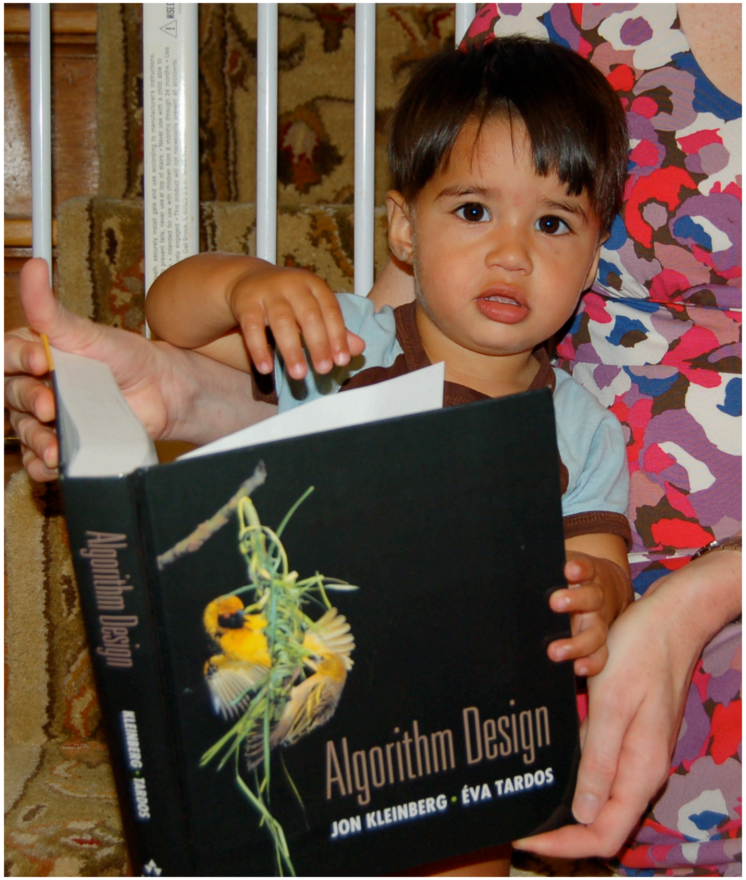
$$T(N) = \max_{\mathbf{x}:\mathbf{x} \text{ is of size } N} t_{\mathcal{A}}(\mathbf{x}).$$

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]

Questions?

Rest of today’s agenda

Analyzing the run time of the GS algo

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
```

Implementation Steps

How do we represent the input?

How do we find a free woman w?

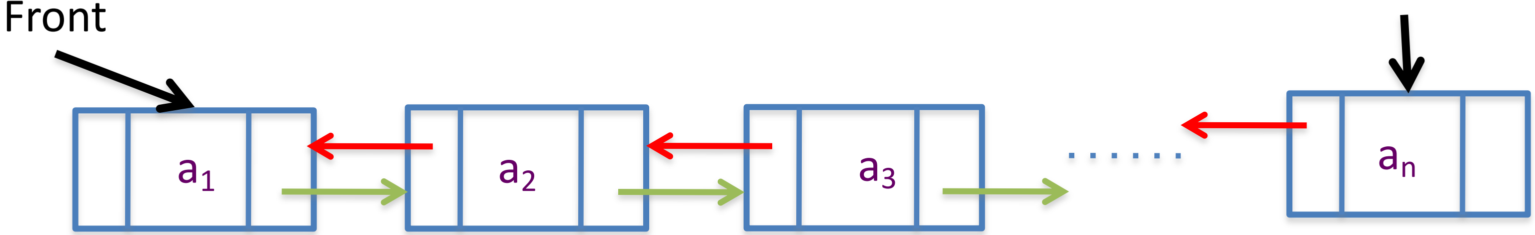
How would w pick her best unproposed man m?

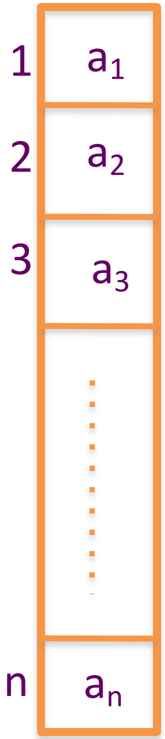
How do we know who m is engaged to?

How do we decide if m prefers w’ to w?

Arrays and Linked Lists

n numbers $a_1,a_2,\dots,a_n$



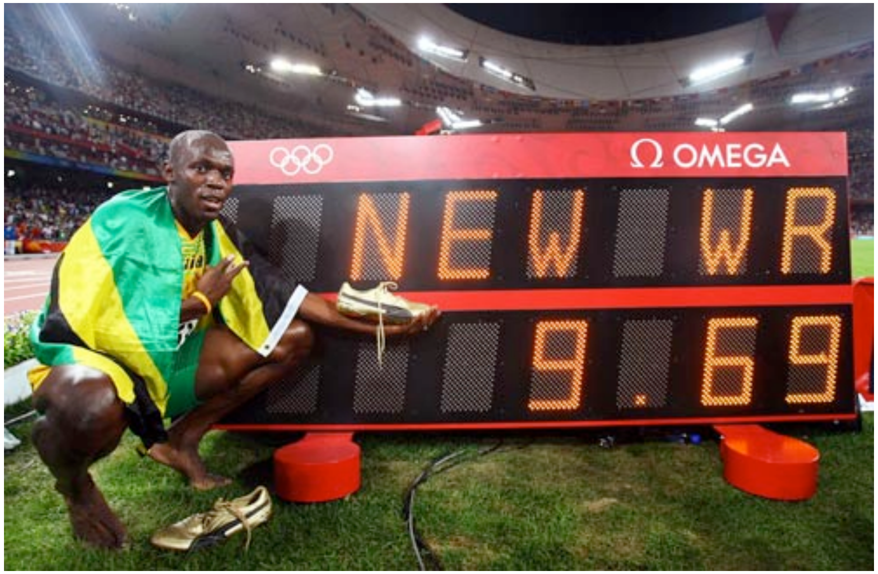


	Array	Linked List
Access ith element	O(1)	O(i)
Is e present?	O(n) (O(log n) if sorted)	O(n)
Insert an element	O(n)	O(1) given pointer
Delete an element	O(n)	O(1) given pointer
Static vs Dynamic	Static	Dynamic

Today's agenda

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

More practice with run time analysis



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
```

At most n^2 iterations

O(1) time implementation