Lecture 7

CSE 331

Sep 13, 2017

Things to remember

Make sure you follow the HW policies

If by chance you violated any, just don't submit

Make sure you clearly demarcate your submission

Use/follow the provided template

Check your PDF submits

Make sure to preview your PDF submission to Autolab!

A corrupted PDF file will get you a zero on that question

Your PDFs cannot be more than 2MB big

GS algo outputs a stable matching

Last lecture, GS outputs a perfect matching S

Lemma 3: S has no instability

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

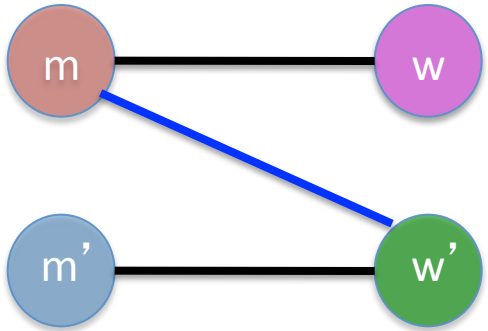
Proof of Lemma 3

By contradiction

Assume there is an instability (m,w')

m prefers w' to w

w' prefers m to m'



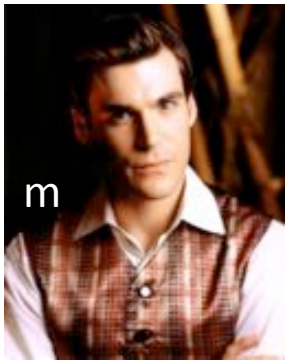
Contradiction by Case Analysis

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’

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

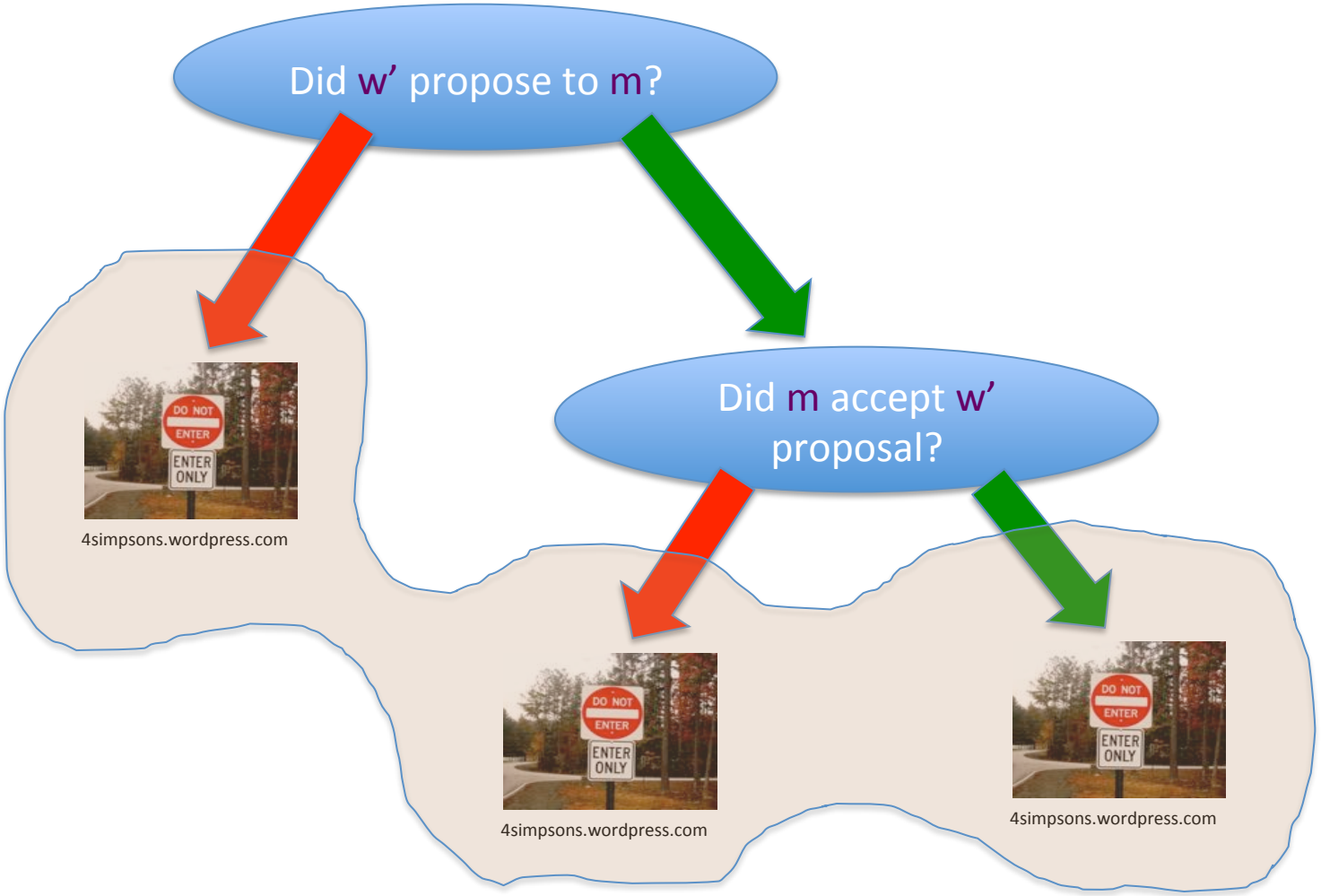
m is finally engaged to w (prefers w to w’’

By Obs 1

m prefers w to w’

4simpsons.wordpress.com

Overall structure of case analysis



Questions?

Extensions

Fairness of the GS algorithm

Different executions of the GS algorithm

Main Steps in Algorithm Design

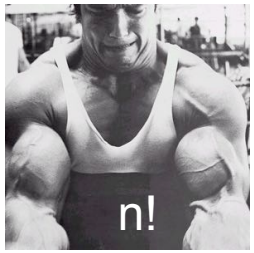
Problem Statement

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?

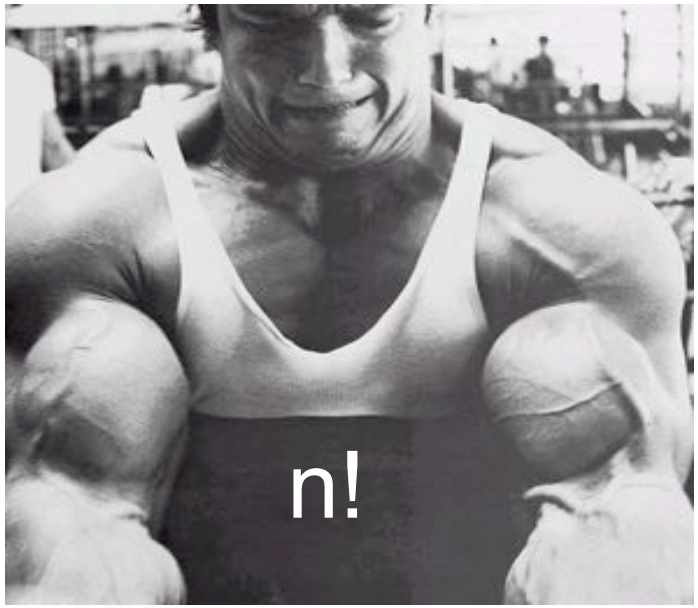
Worst-case Inputs

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)

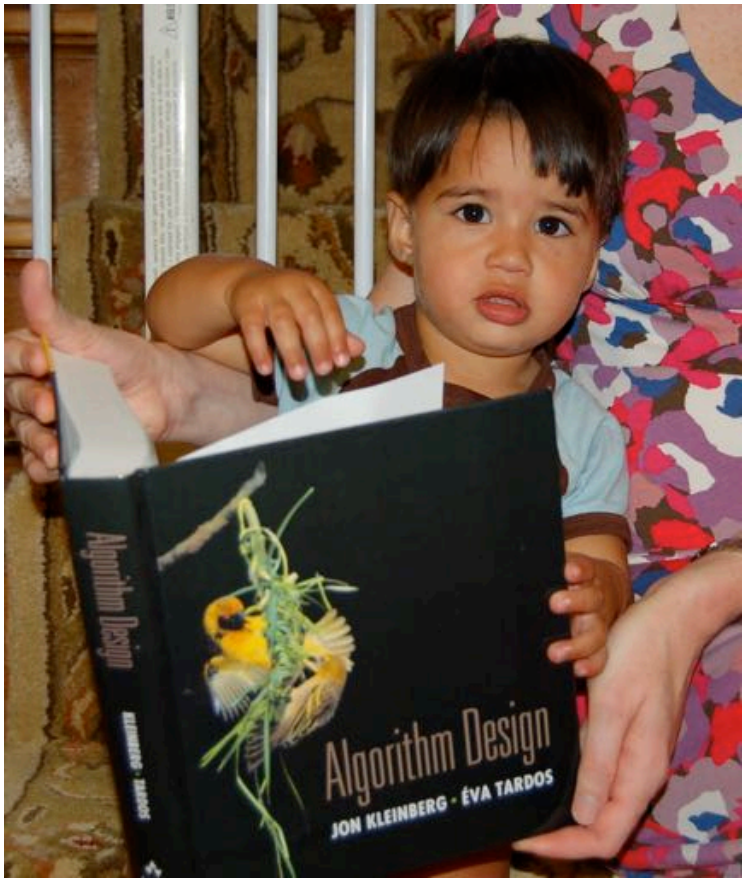
Step: “primitive computational step”

More on polynomial time

Problem centric tractability

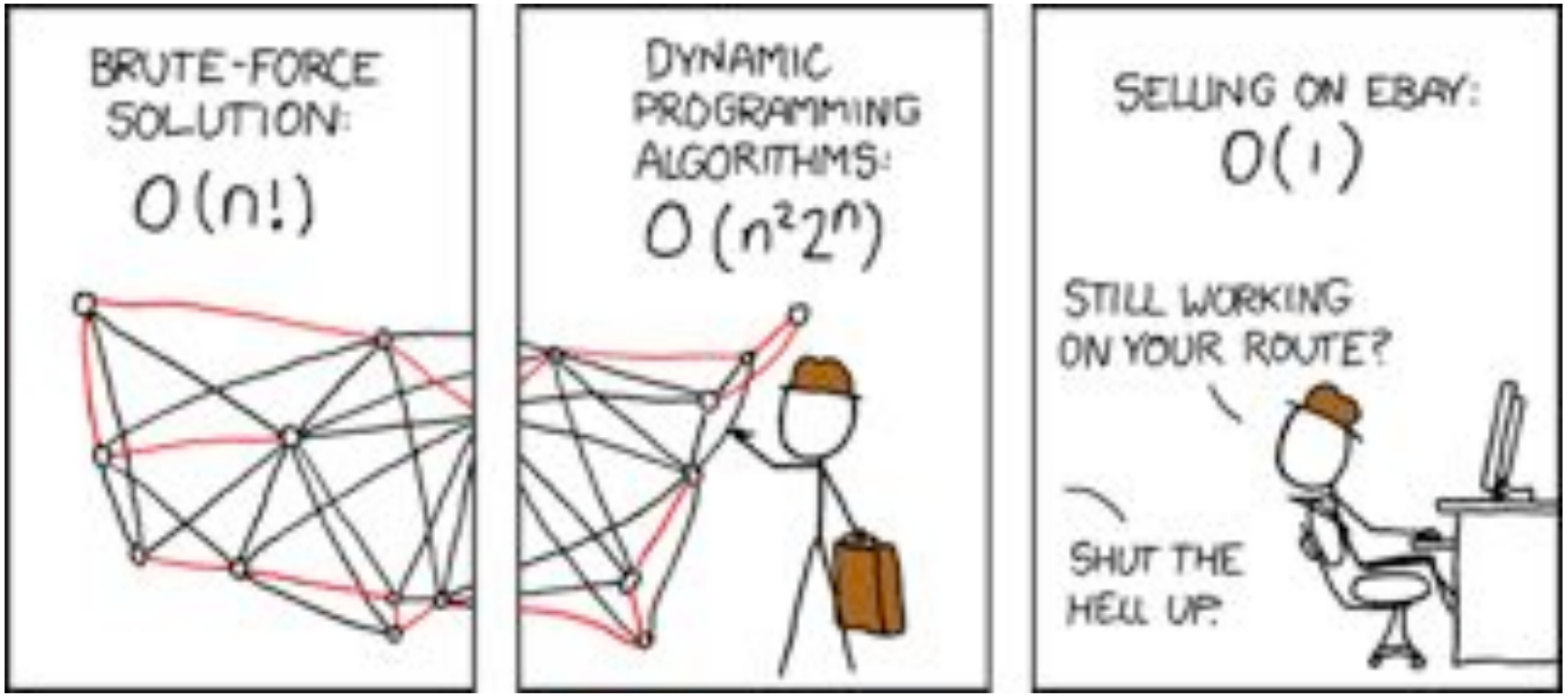
Can talk about problems that are not efficient!

Reading Assignments



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

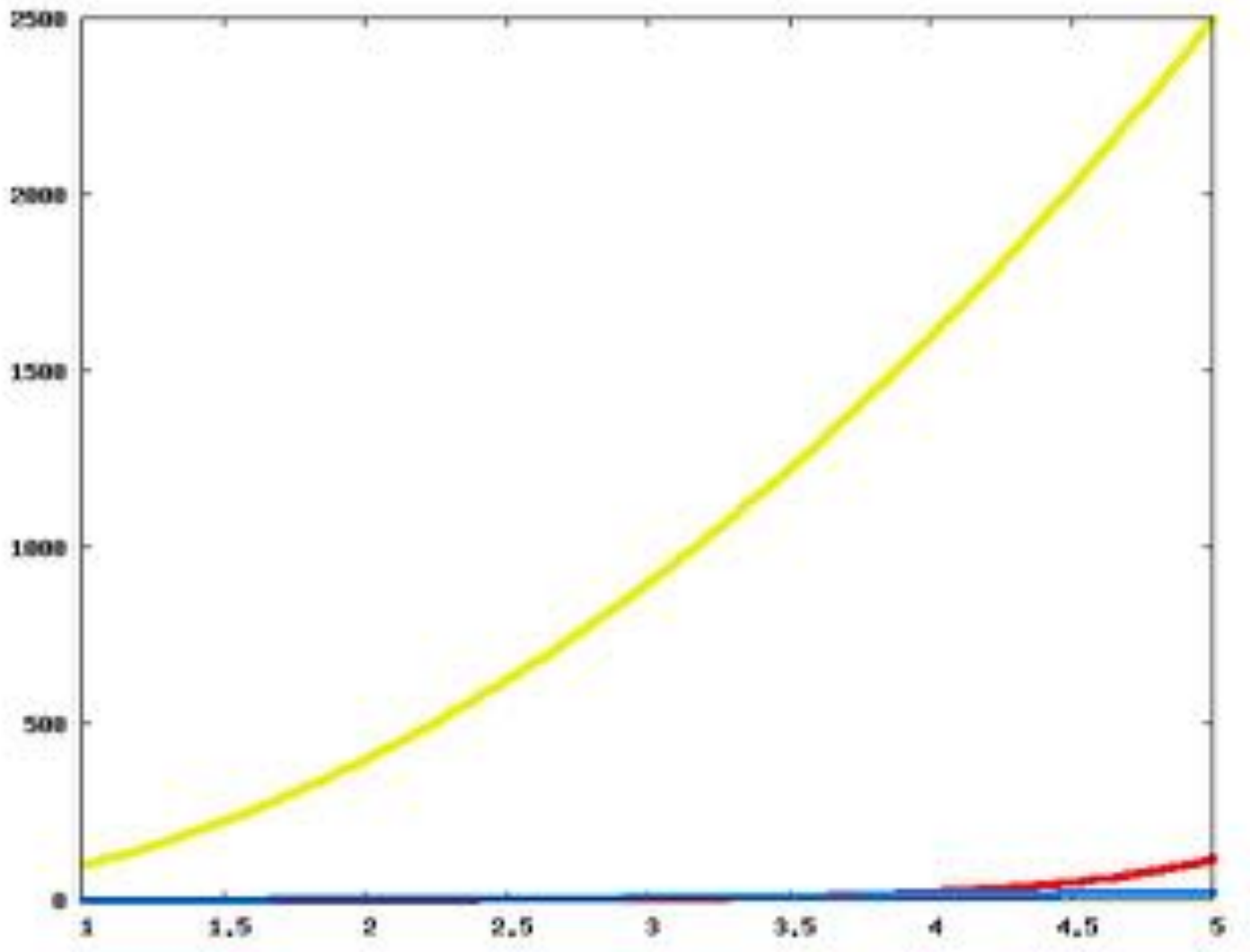
Asymptotic Analysis



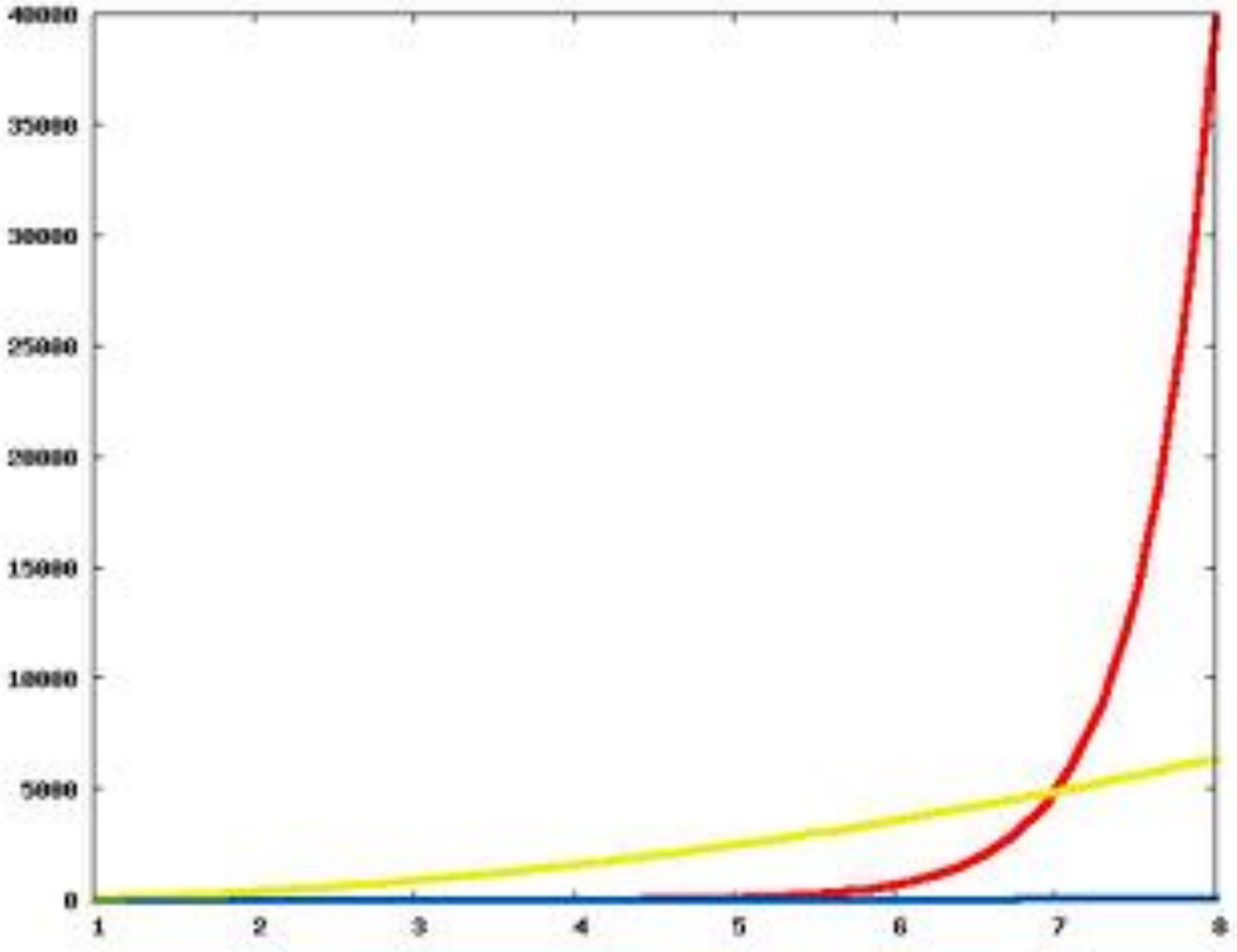
Travelling Salesman Problem

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

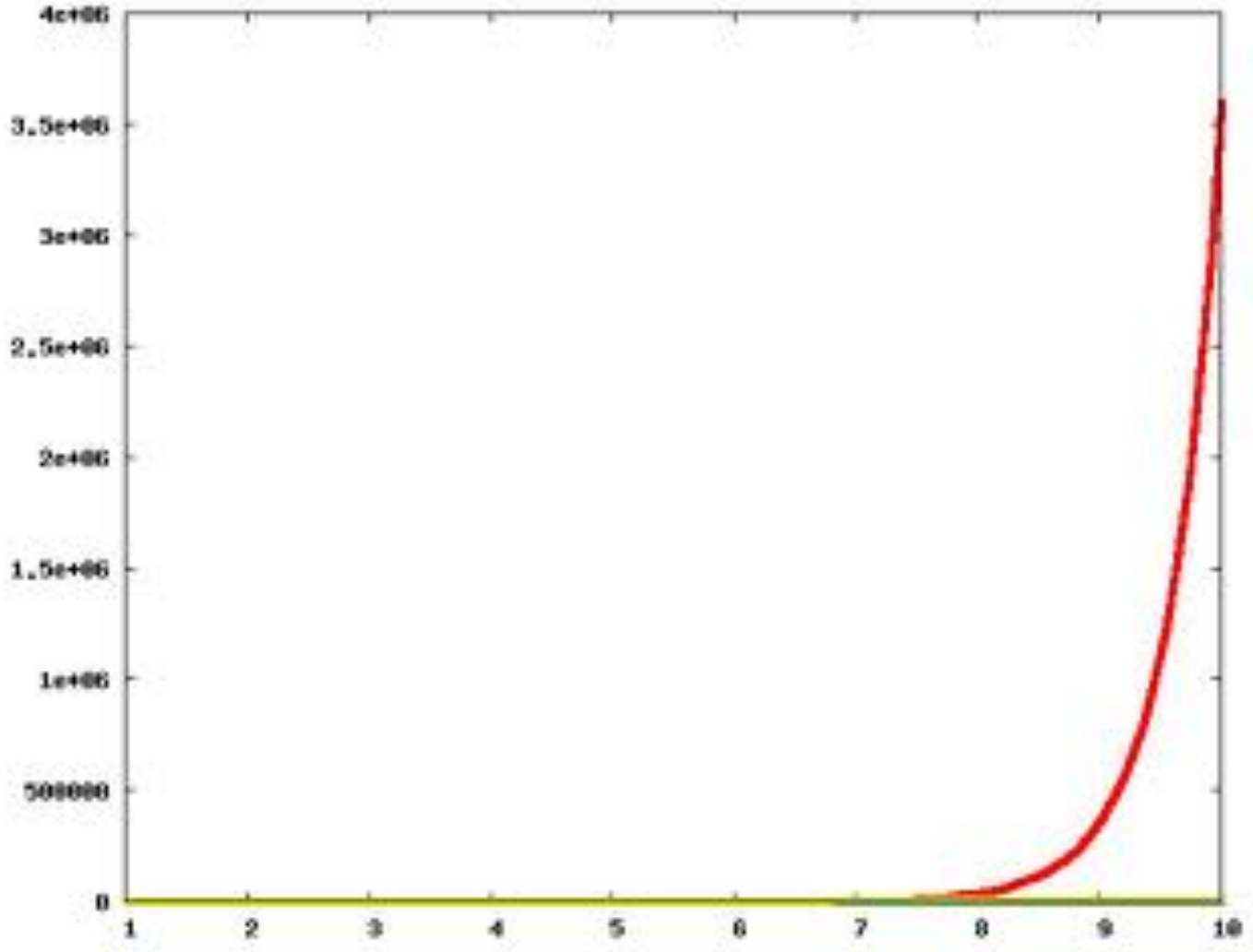
Which one is better?



Now?



And now?

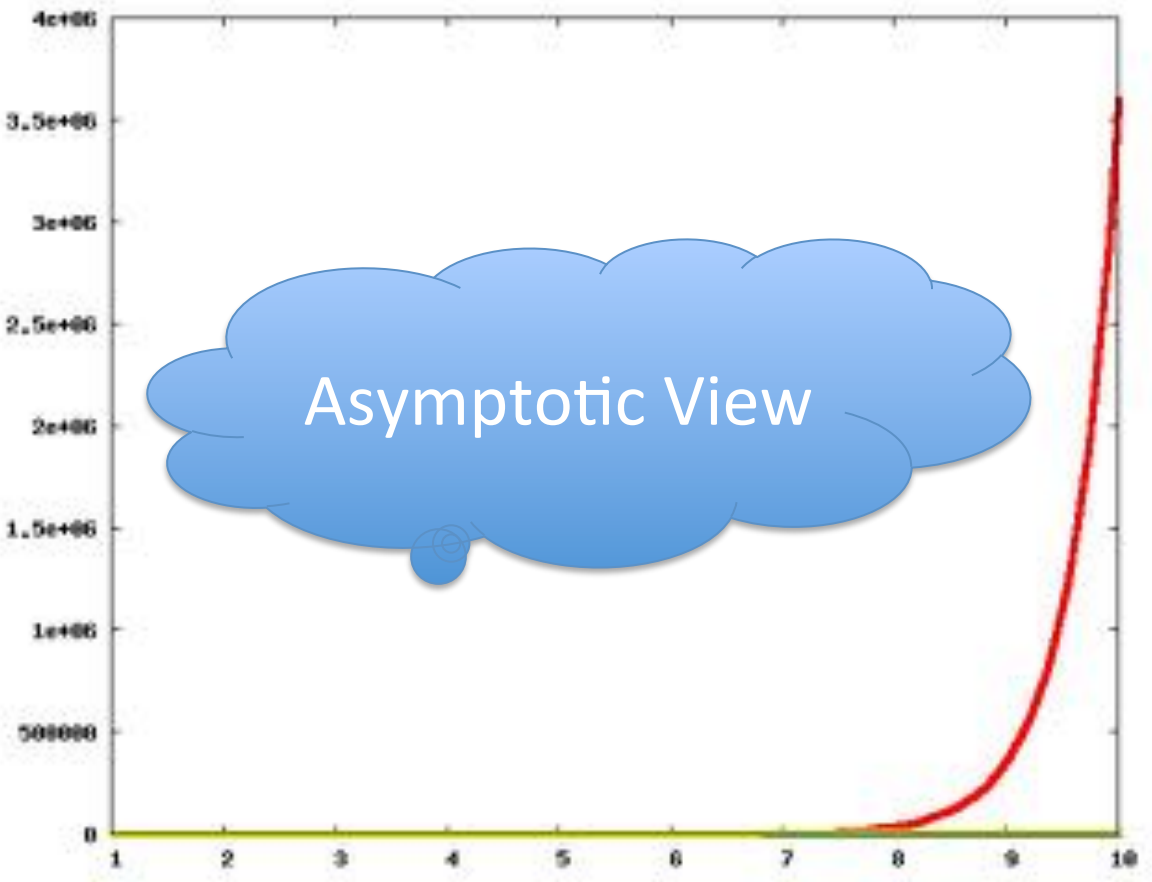


The actual run times

n!

$100n^2$

n^2



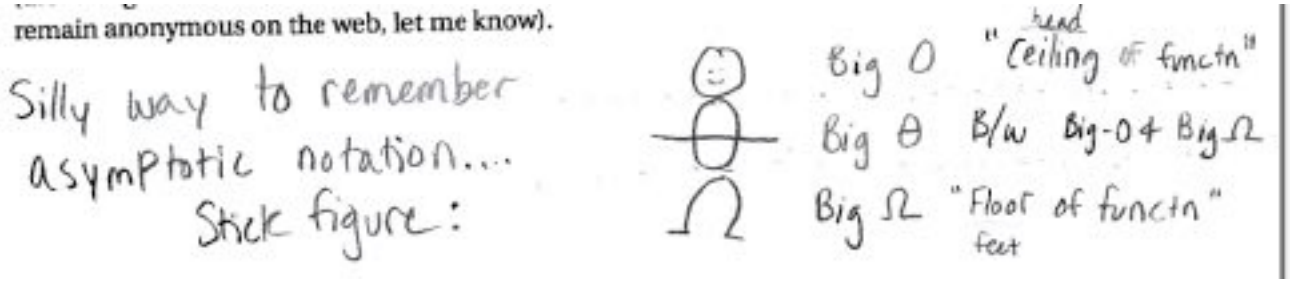
Asymptotic Notation

≤ is O with glasses

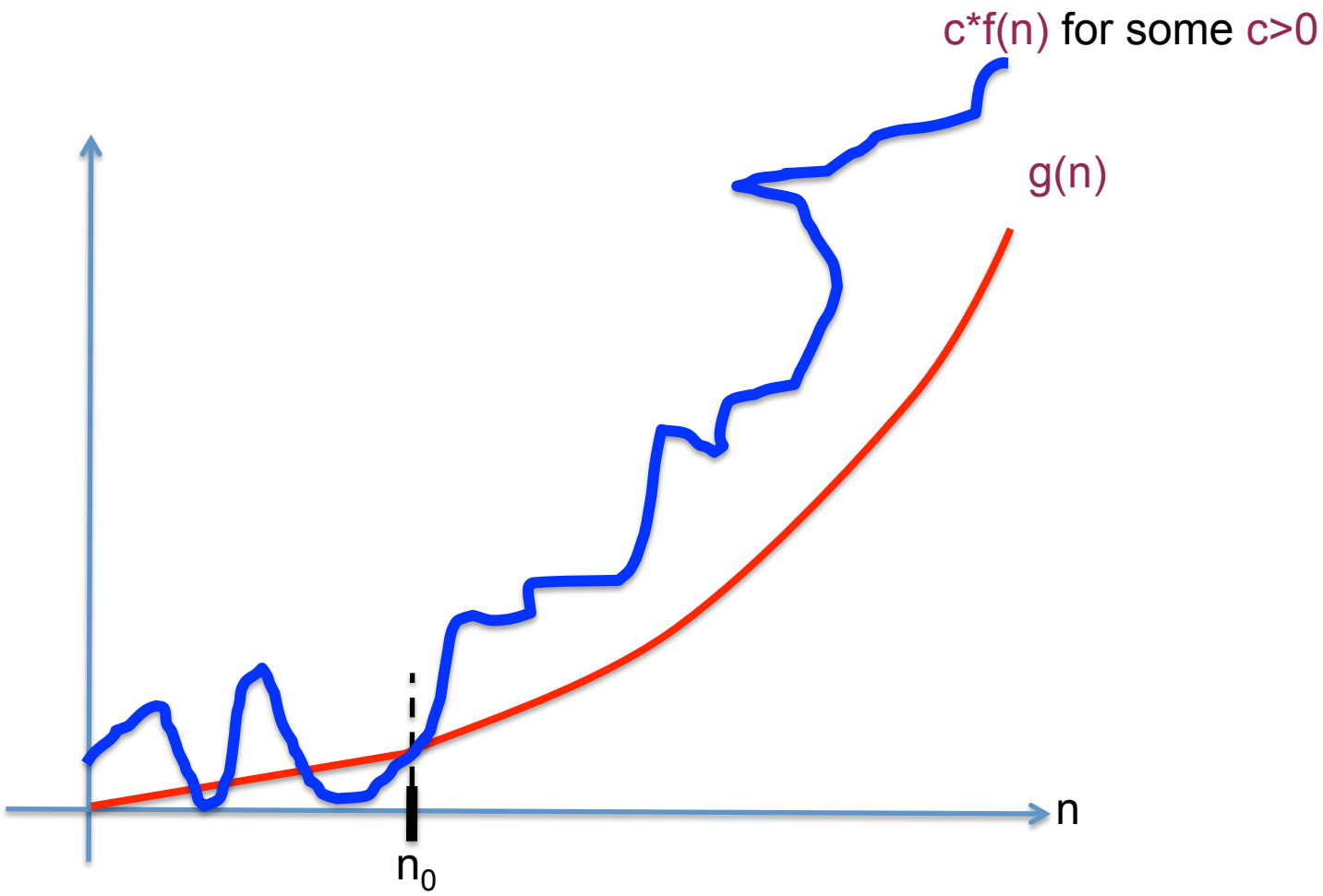
≥ is Ω with glasses

= is Θ with glasses

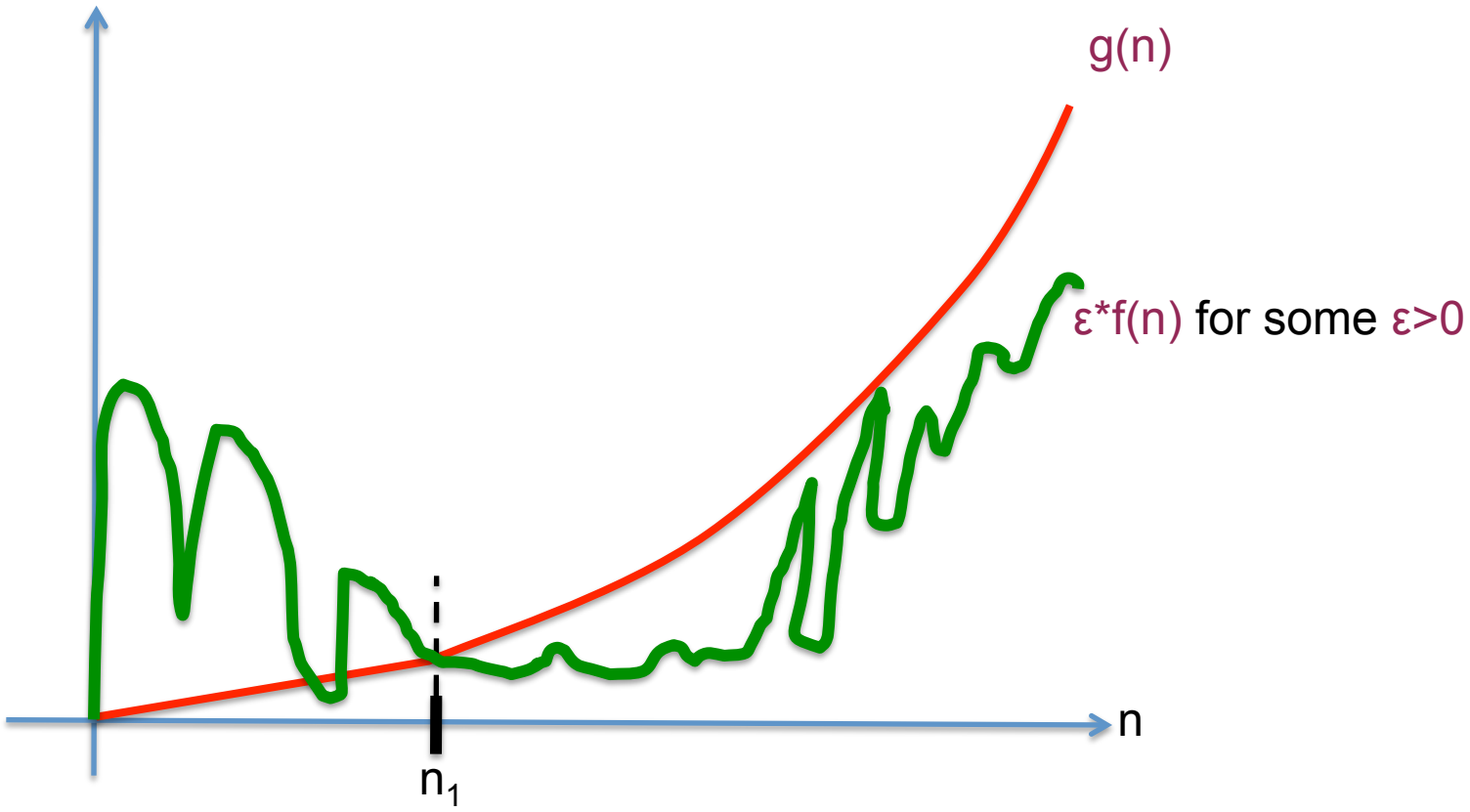
Another view



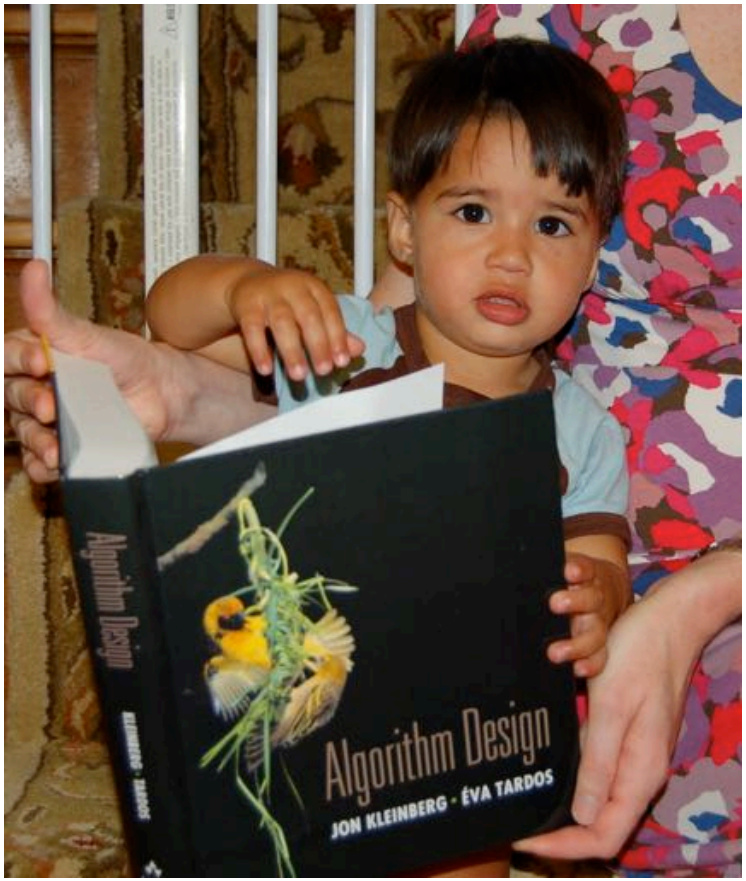
g(n) is O(f(n))



g(n) is Ω(f(n))



Reading Assignments



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