Lecture 8

CSE 331

Sep 14, 2018

HW 2 has been posted

Homework 2

Due by **11:59pm, Thursday, September 20, 2018**.

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

Project group due in a week

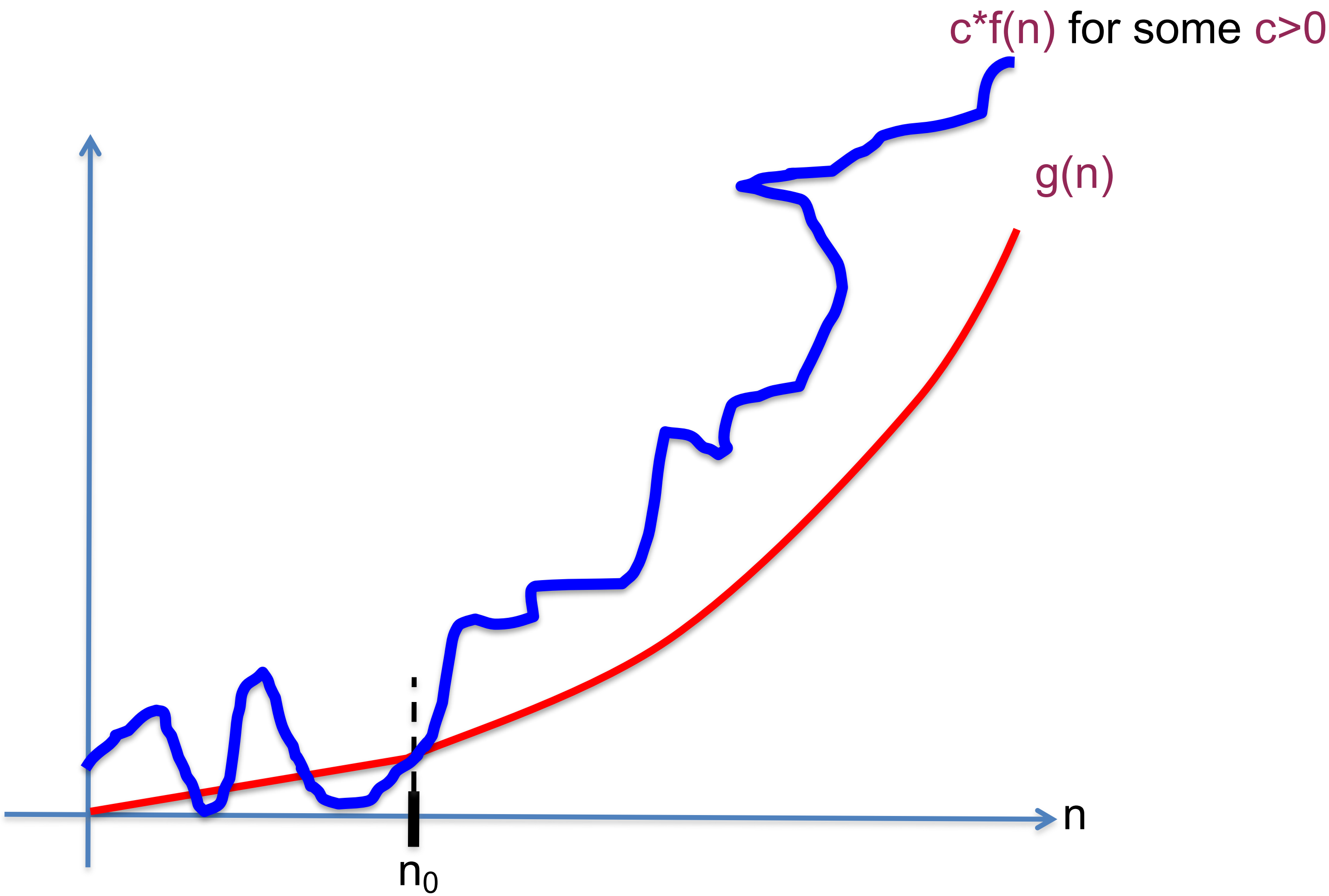
CSE 331 Mini project choices

Fall 2018

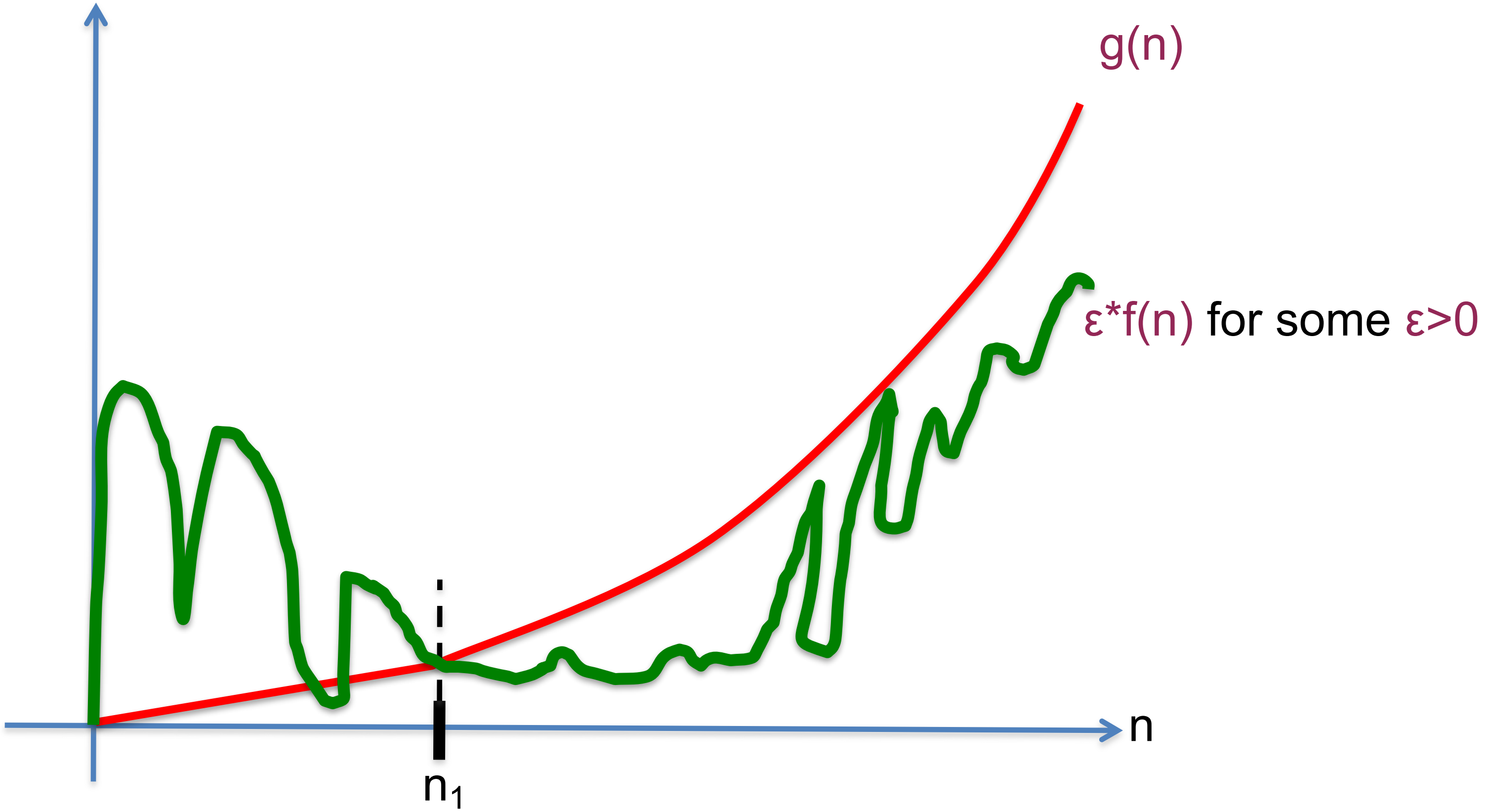
Please check the table below before submitting your mini project team composition to make sure your case study is not being used by another group. Case studies are assigned on a first come first serve basis.

Group	Chosen Algorithm	Case Study	Links
Chinmayee Bandal, Sarah Peters, Tracy Zheng ()	Dijkstra's Algorithm	Google Maps	Link 1, Link 2
Jonathan Wong, Jacky Eng, Jack Bett (Segmentation Fault)	Linkedin Feed Algorithm	Linkedin Feed Relevance System	Link 1, Link 2, Link 3
Waiwai Kim, John Demetros, Frank Tsai (Autonomous Vehicle)	Deep Deterministic Policy Gradients	Reinforcement Learning in Autonomous Vehicle	Link 1, Link 2, Link 3
Matthew Pan, kelvin hong wen wong, Bohang Hua (331MKB)	RSA Algorithm**ALREADY TAKEN-- PLEASE CHOOSE ANOTHER CASE STUDY**	online transaction	Link 1, Link 2, Link 3
Mohammed Shmsuddin, Vencent Feng, Krapi Vani	Data Compression Algorithms	Sound data compression	Link 1

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



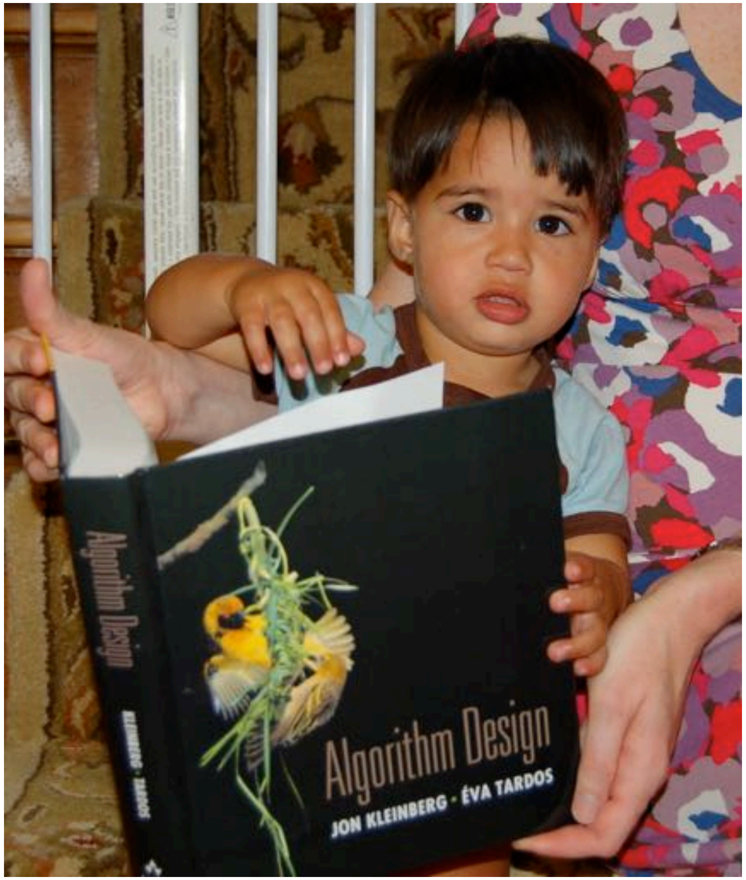
g(n) is Ω(f(n))



Properties of O (and Ω)

Transitive	g is O(f) and f is O(h) then g is O(h)
Additive	g is O(h) and f is O(h) then g+f is O(h)
Multiplicative	g is $O(h_1)$ and f is $O(h_2)$ then g*f is $O(h_1*h_2)$

Reading Assignments



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

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).

Questions?

Today's agenda

Asymptotic run time

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?

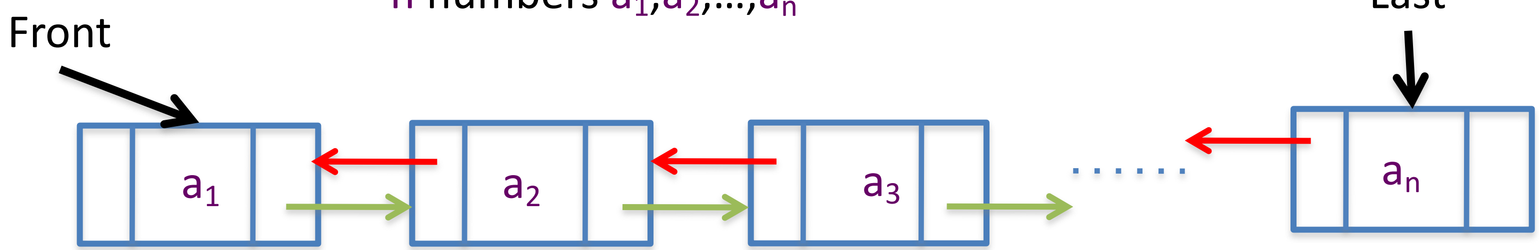
How would w pick her best unproposed man m?

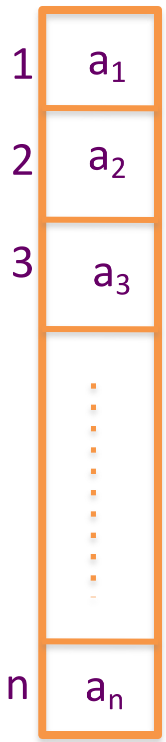
How do we know who m is engaged to?

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

Arrays and Linked Lists

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





	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

At most n^2 iterations

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

O(1) time implementation

Output the engaged pairs as the final output