

# Lecture 9

CSE 331

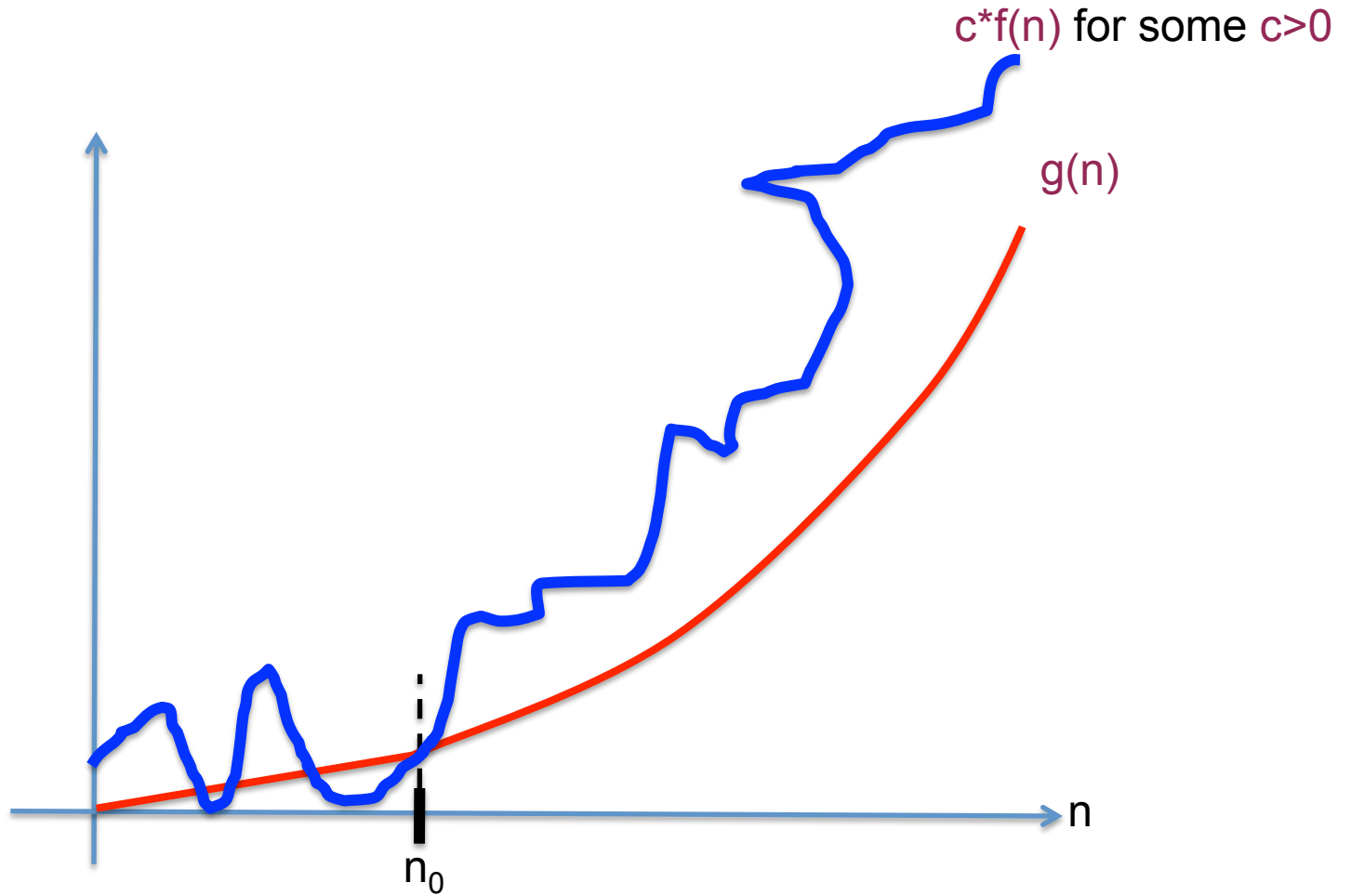
Sep 15, 2014

# Run time of an algorithm

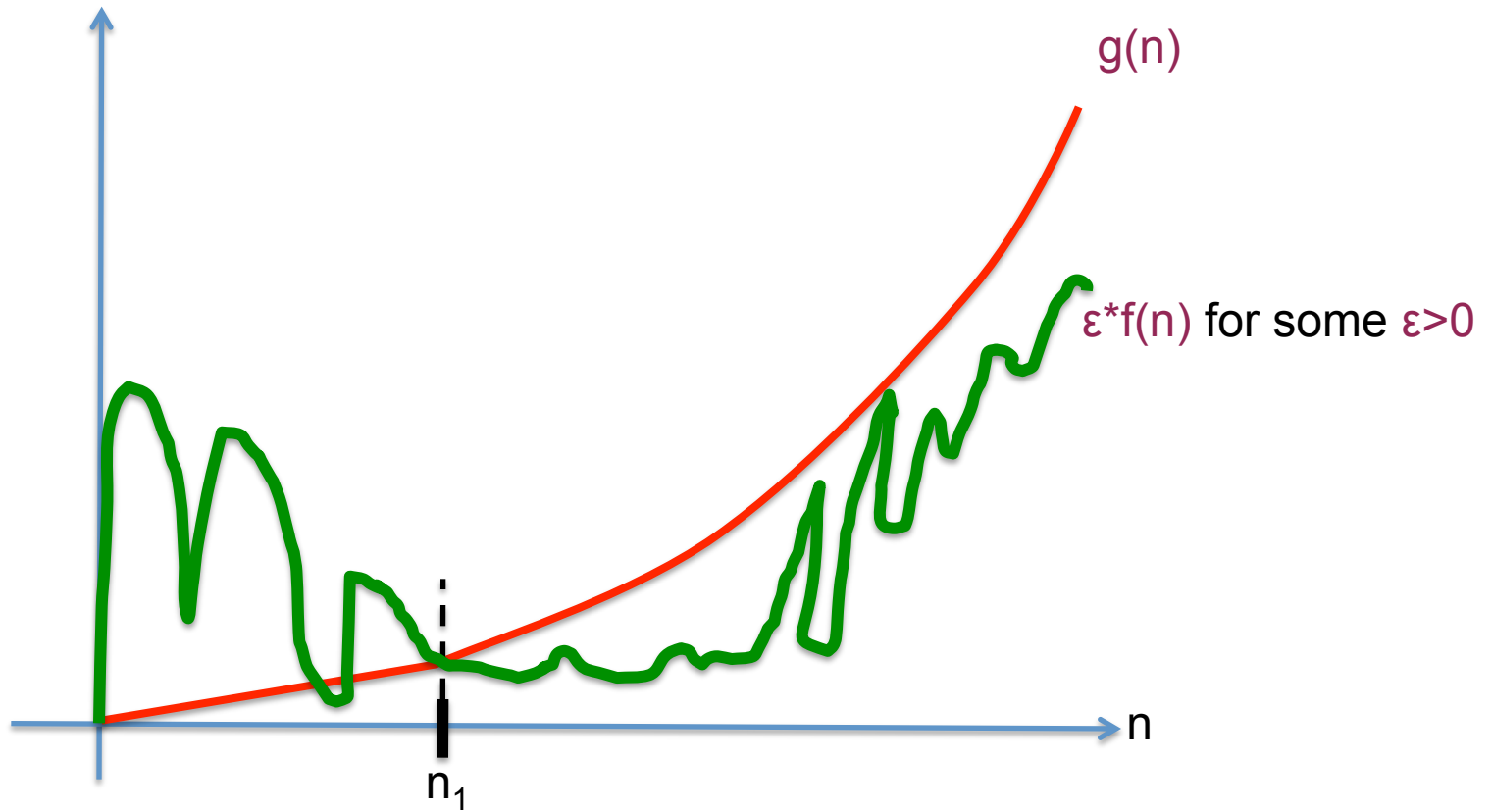
(Worst-case) run time  $T(n)$  for input size  $n$

Maximum number of steps taken by the algorithm for **any** input of size  $n$

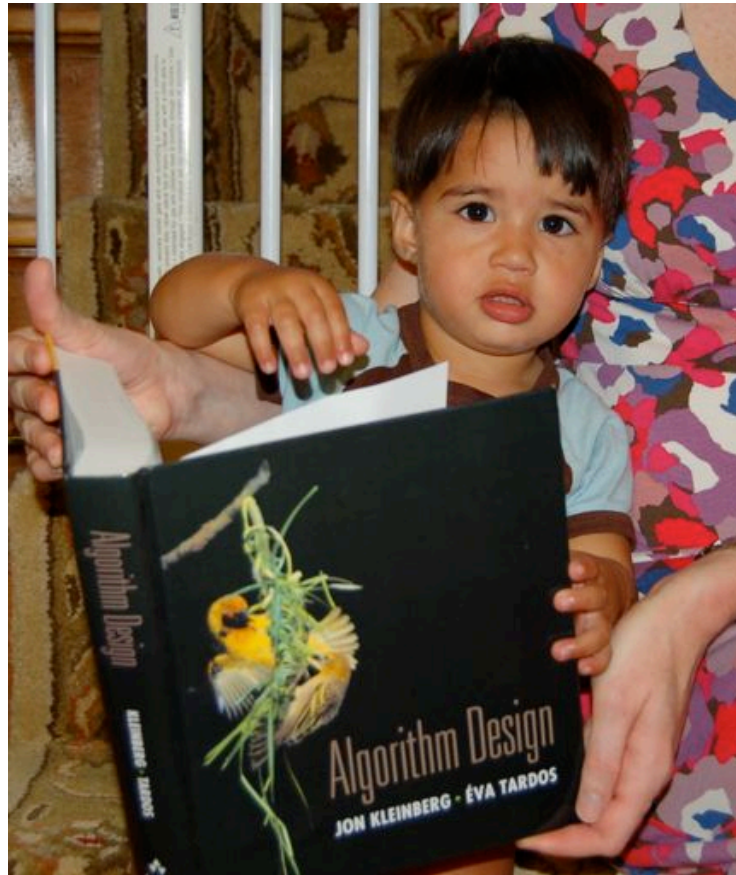
$g(n)$  is  $O(f(n))$



$g(n)$  is  $\Omega(f(n))$



# Reading Assignments



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

# Questions?



# Today's agenda

Asymptotic run time

Analyzing the run time of the GS algo

# Gale-Shapley Algorithm

Initially 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?

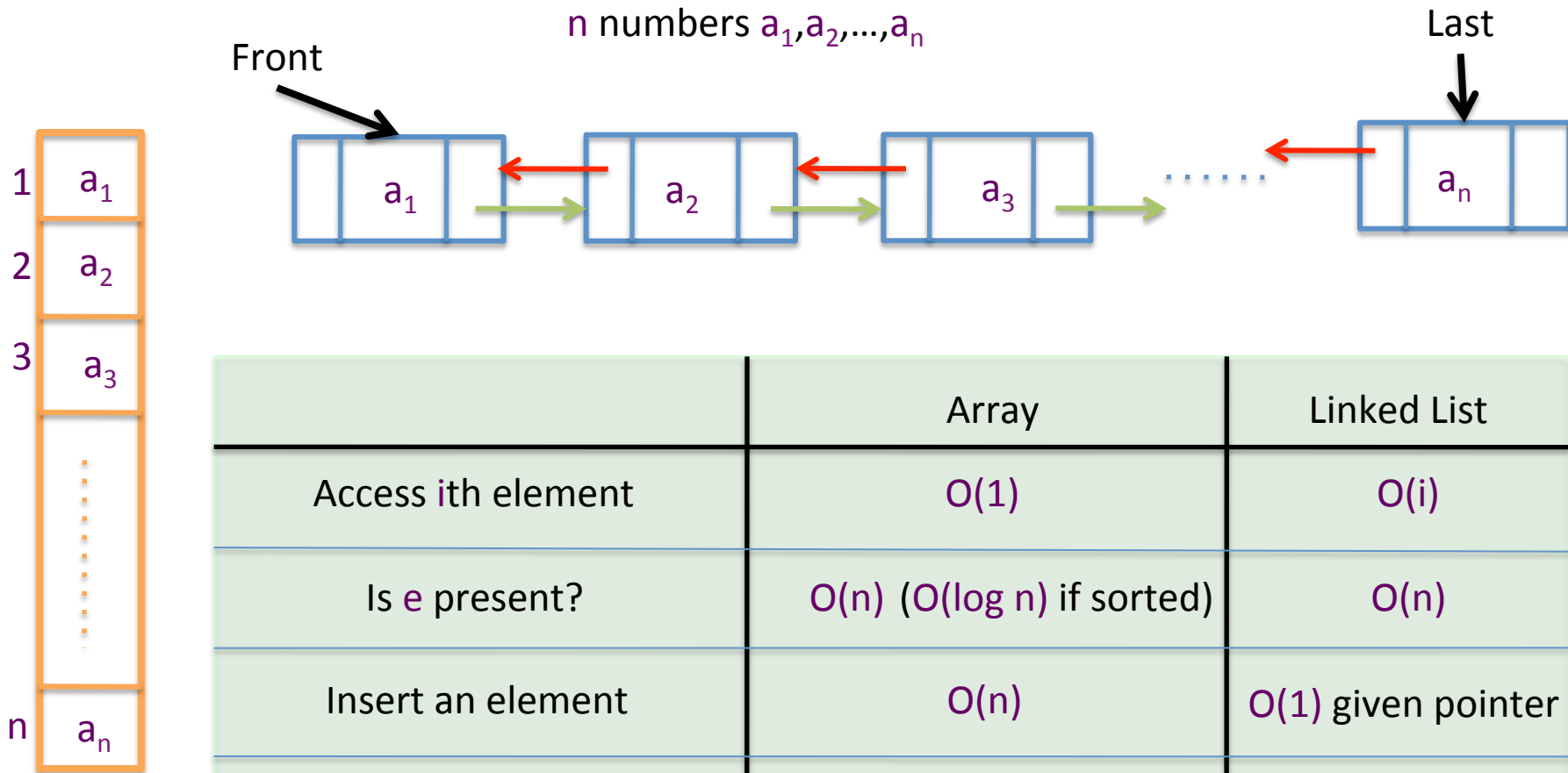
How do we find a free woman  $w$ ?

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



	Array	Linked List
Access $i$ th 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

Initially all men and women are **free**

At most  $n^2$  iterations

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