

# CS61A Discussion 7: **Orders of Growth and Trees**

TA: **Jerry Chen**

Email: **[jerry.c@berkeley.edu](mailto:jerry.c@berkeley.edu)**

TA Website: **[jerryjrchen.com/cs61a](http://jerryjrchen.com/cs61a)**

# Attendance

Form: **[tinyurl.com/jerrydisc](https://tinyurl.com/jerrydisc)**

**Please remember to checkin! Only ~17 from last week**

For the weekly question,  
**please tell me your least favorite part of hw04.**

(Of course, please only check in if you showed up!)

# Agenda

1. Week in Review
2. Composition
3. Orders of Growth
4. Trees (with mutation)

# Week In Review

Hw4

- Was challenging!

Ants - Due next Thursday

- It's ok, I haven't started either :)

Lab 7 (Recursive Objects) - Due Friday

Mt2 - **7-9pm, Wednesday** after Spring Break (3/30)

- ***Submit alternate time request ASAP!***

Hog composition

# Composition

**Computers don't care** how “neat” your code is

**Humans** (coworkers, boss, future you?) **do care!**

```
1 @font-face{font-family:'footable';src:url('fonts/footable.eot');src:url('fonts/footable.eot?#iefix') format('embedded-opentype'),url('fonts/footable.woff') format('woff'),url('f
onts/footable.ttf') format('truetype'),url('fonts/footable.svg#footable') format('svg');font-weight:normal;font-style:normal}@media screen and (-webkit-min-device-pixel-ratio:0)
{@font-face{font-family:'footable';src:url('fonts/footable.svg#footable') format('svg');font-weight:normal;font-style:normal}}.footable{width:100%.footable.breakpoint>tbody>tr.
footable-detail-show>td{border-bottom:0}.footable.breakpoint>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e001"}.footable.breakpoint>tbody>tr:ho
ver:not(.footable-row-detail){cursor:pointer}.footable.breakpoint>tbody>tr>td.footable-cell-detail{background:#eee;border-top:0}.footable.breakpoint>tbody>tr>td>span.footable-to
ggle{display:inline-block;font-family:'footable';speak:none;font-style:normal;font-weight:normal;font-variant:normal;text-transform:none;-webkit-font-smoothing:antialiased;padding-ri
ght:5px;font-size:14px;color:#888}.footable.breakpoint>tbody>tr>td>span.footable-toggle:before{content:"\e000"}.footable.breakpoint.toggle-circle>tbody>tr.footable-detail-show>td>
span.footable-toggle:before{content:"\e005"}.footable.breakpoint.toggle-circle>tbody>tr>td>span.footable-toggle:before{content:"\e004"}.footable.breakpoint.toggle-circle-filled>
tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e003"}.footable.breakpoint.toggle-circle-filled>tbody>tr>td>span.footable-toggle:before{content:"\e002"}.f
ootable.breakpoint.toggle-square>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e007"}.footable.breakpoint.toggle-square>tbody>tr>td>span.footable-to
ggle:before{content:"\e006"}.footable.breakpoint.toggle-square-filled>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e009"}.footable.breakpoint.toggle-square
-filled>tbody>tr>td>span.footable-toggle:before{content:"\e008"}.footable.breakpoint.toggle-arrow>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e00f"}.f
ootable.breakpoint.toggle-arrow>tbody>tr>td>span.footable-toggle:before{content:"\e011"}.footable.breakpoint.toggle-arrow-small>tbody>tr.footable-detail-show>td>span.footable-to
ggle:before{content:"\e013"}.footable.breakpoint.toggle-arrow-small>tbody>tr>td>span.footable-toggle:before{content:"\e015"}.footable.breakpoint.toggle-arrow-circle>tbody>tr.fo
otable-detail-show>td>span.footable-toggle:before{content:"\e01b"}.footable.breakpoint.toggle-arrow-circle>tbody>tr>td>span.footable-toggle:before{content:"\e01d"}.footable.brea
kpoint.toggle-arrow-circle-filled>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e00b"}.footable.breakpoint.toggle-arrow-circle-filled>tbody>tr>td>span.fo
otable-toggle:before{content:"\e00d"}.footable.breakpoint.toggle-arrow-tiny>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content:"\e01f"}.footable.breakpoint.tog
gle-arrow-tiny>tbody>tr>td>span.footable-toggle:before{content:"\e021"}.footable.breakpoint.toggle-arrow-alt>tbody>tr.footable-detail-show>td>span.footable-toggle:before{content
:"\e017"}.footable.breakpoint.toggle-arrow-alt>tbody>tr>td>span.footable-toggle:before{content:"\e019"}.footable.breakpoint.toggle-medium>tbody>tr>td>span.footable-toggle{font-s
ize:18px}.footable.breakpoint.toggle-large>tbody>tr>td>span.footable-toggle{font-size:24px}.footable>thead>tr>th{-webkit-touch-callout:none;-webkit-user-select:none;-khtml-user
-select:none;-moz-user-select:-moz-none;-ms-user-select:none;user-select:none}.footable>thead>tr>th.footable-sortable:ho
ver{cursor:pointer}.footable>thead>tr>th.footable-sorted>span.footable-sort-indicator:before{content:"\e013"}.footable>thead>tr>th.footable-sorted-desc>span.footable-sort-indicator:
before{content:"\e012"}.footable>thead>tr>th>span.footable-sort-indicator{display:inline-block;font-family:'footable';speak:none;font-style:normal;font-weight:normal;font-variant:
normal;text-transform:none;-webkit-font-smoothing:antialiased;padding-left:5px}.footable>thead>tr>th>span.footable-sort-indicator:before{content:"\e022"}.footable>tfoot .pagination{margin:0}.footable.no-paging .hide-if-no-paging{display:none}.footable-row-detail-inner{display:table}.footable-row-detail-row{display:table-row;line-height:1.5em}.footable-row-detail-group{display:block;line-height:2em;font-size:1.2em;font-weight:bold}.footable-row-detail-name{display:table-cell;font-weight:bold;padding-right:.5em}.footable-row-detail-value{display:table-cell}.footable-odd{background-color:#f7f7f7}/*!normalize.css v1.1.3 | MIT License | git.io/normalize */ article aside details figcaption figure footer header hgroup main nav section summary{display:block}
```

# Composition

## **Basic Principles**

- Pretend you are reading the code for the first time
- Simple > Complex
- Pick meaningful names
- (Personal opinion) give operators room to breathe!

# Composition

**Composition is not as important as correctness.**

In my opinion:

- Correctness > Efficiency > Composition

“Done is better than perfect” - Facebook

Composition & style guide are at: <http://cs61a.org/articles/resources.html>

# Orders of Growth

Context change — from writing programs to **evaluating their performance**

How do we describe how fast a program is?



# Orders of Growth

Why do we care?

In the news



**Google's DeepMind defeats legendary Go player Lee Se-dol in historic victory**

**The Verge** - 1 day ago

DeepMind founder Demis Hassabis expressed "huge respect for Lee Se-dol and his ...

**Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo**

**YouTube** - 1 day ago

**Google's Deepmind AI beats Go world champion in first match**

**Engadget** - 23 hours ago

**More news for deepmind**

# Orders of Growth

One way is through orders of growth. How does a program respond to a growing input size?

$\Omega$

$\Theta$

$O$

# Orders of Growth

```
1 def factorial(n):  
2     if n == 0:  
3         return 1  
4     return n * factorial(n - 1)
```

How does this scale with respect to size of input  $n$ ?

Linear -  $O(n)$

# Orders of Growth

Growth Rate (Big-O)	How does it feel?
1 (“constant”)	Great!
$\log N$ (“logarithmic”)	Still really good!
$N$ (“linear”)	Not bad
$N^2, N^3$ (“poly time”)	Acceptable
$2^N$ (“exponential”)	Ugh... “intractable” growth
$N!$ (“evil” “factorial”)	Same as above... but worse?



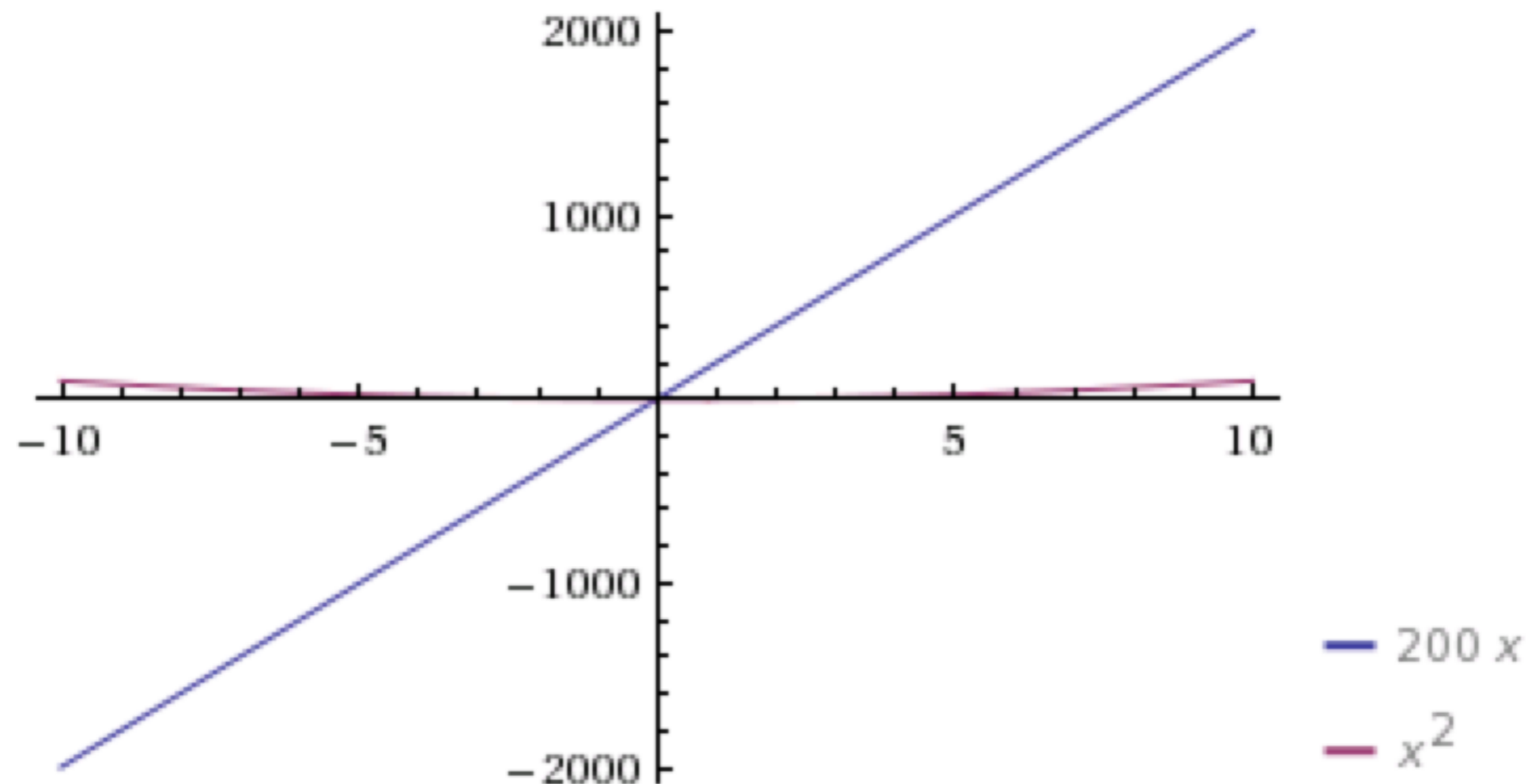
# Orders of Growth

Time ( $\mu\text{sec}$ ) for problem size $N$	Max $N$ Possible in			
	1 second	1 hour	1 month	1 century
$\lg N$	$10^{300000}$	$10^{10000000000}$	$10^{8 \cdot 10^{11}}$	$10^{9 \cdot 10^{14}}$
$N$	$10^6$	$3.6 \cdot 10^9$	$2.7 \cdot 10^{12}$	$3.2 \cdot 10^{15}$
$N \lg N$	63000	$1.3 \cdot 10^8$	$7.4 \cdot 10^{10}$	$6.9 \cdot 10^{13}$
$N^2$	1000	60000	$1.6 \cdot 10^6$	$5.6 \cdot 10^7$
$N^3$	100	1500	14000	150000
$2^N$	20	32	41	51

# Orders of Growth

Fantastic tool, but it has limitations:

Plot:

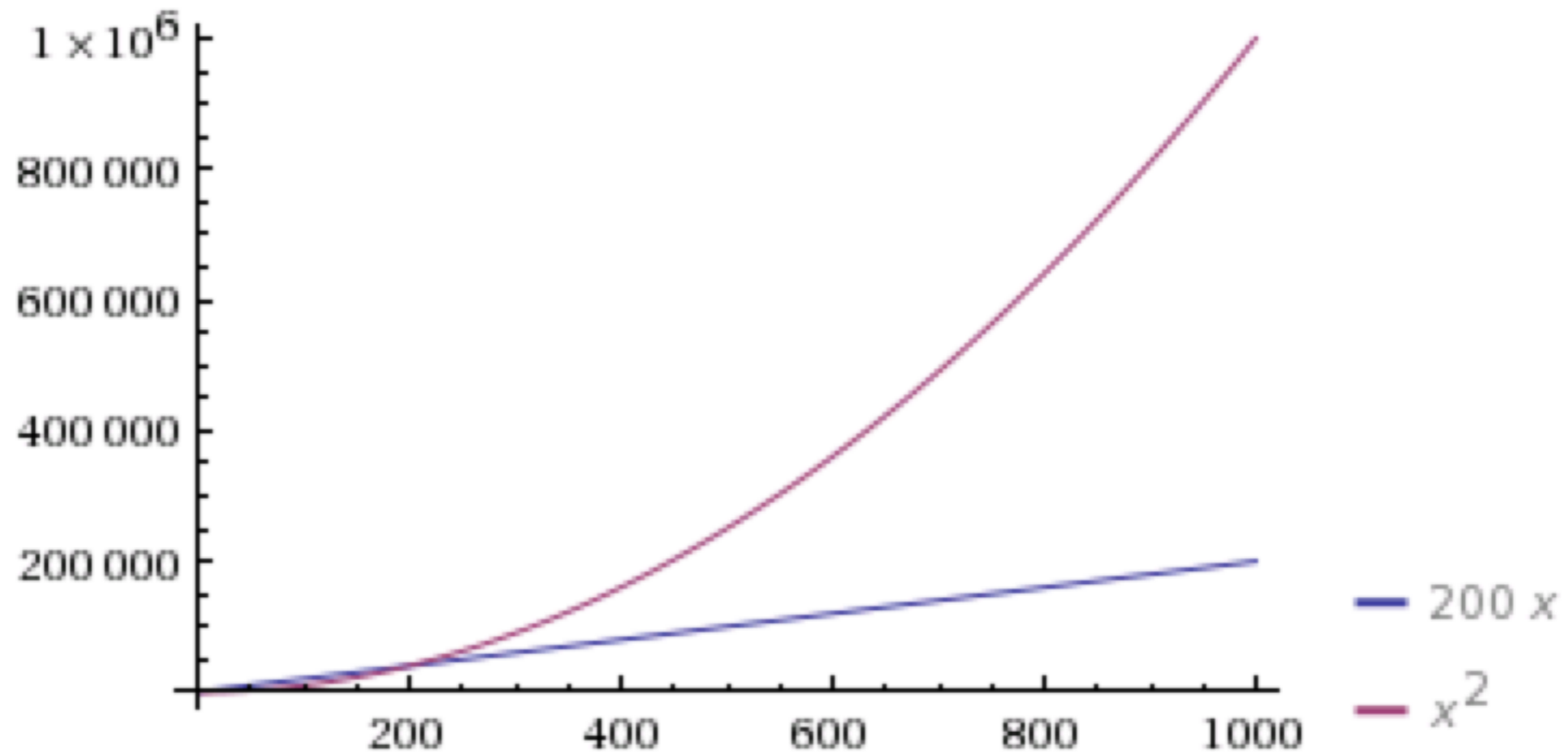


Graphs generated by WolframAlpha

# Orders of Growth

But Big-O still wins in the end!

Plot:





# Orders of Growth

**Simplify**

**Answer**

$O(3n)$

$O(n)$  — ignore const factors

$O(n^3 + 1000n^2)$

$O(n^3)$  — larger term dominates

$O(\log n + n)$

$O(n)$  — larger term dominates

$O(n \log n + n)$

$O(n \log n)$  — larger term dominates

\*\*Caveat — **these are NOT mathematically precise** ways of describing growth relationships!

# Orders of Growth

Question\*\*

Answer

Is factorial  $O(n!)$ ?  $O(\log n)$ ?

Yes, No.  $n!$  greatly upper bounds it.  $\log n$  is not a sufficient upper bound.

$O(\log_2 n) > O(\log_{10} n)$

No! Use change of base formula.

$O(n \log(n^8)) > O(n^2 \log(n^3))$

No — use log rules to get  $O(n \log n)$  vs  $O(n^2 \log n)$

$O(n \log n) < O(\log n^{\log n})$

Yes — RHS is  $n^{\log \log n}$  (try to introduce an exponent)

# Mutable Trees



# Mutable Trees

Can still build trees in much the same way:

```
Tree(<label>, [Tree(...), Tree(...), ...])
```

Selectors now also allow assignment:

```
t = Tree(1)
```

```
t.label = 10
```

# Mutable Trees

Interface:

```
Tree(label, [list_of_children])
```

```
Tree.label
```

```
Tree.branches (or Tree.children)
```

```
Tree.is_leaf(self)
```