Discussion 07:
Orders of Growth and Trees

....

TA: **Jerry Chen**
Email: **jerry.c@berkeley.edu**
TA Website: **jerryjrchen.com/cs61a**

Agenda

1. Orders of Growth (OOG)

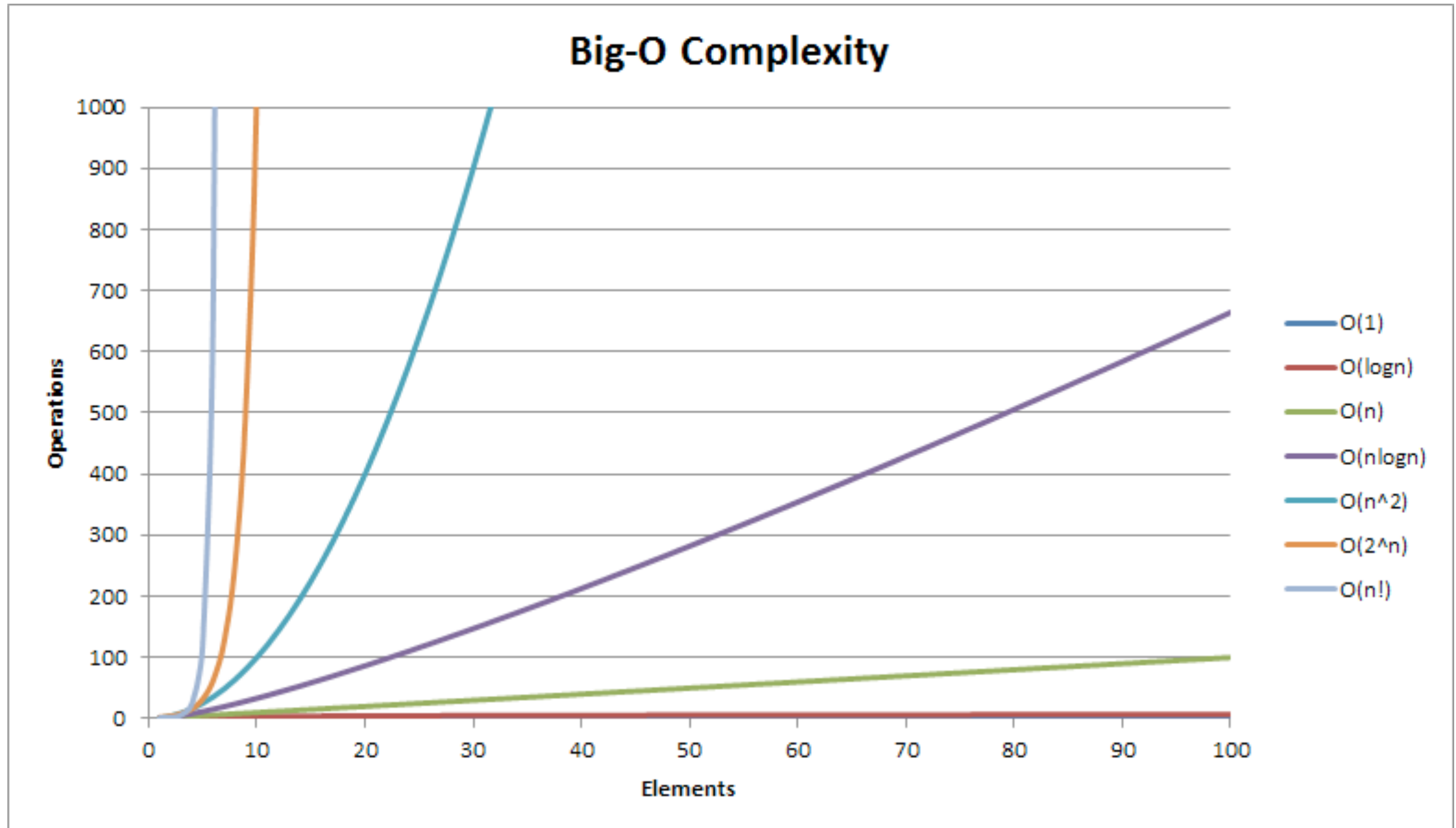2. Trees (skip slides)

Announcements

Midterm 2 **Wed 3/15 8-10pm**. Conflict form emailed out, fill out by this Friday!

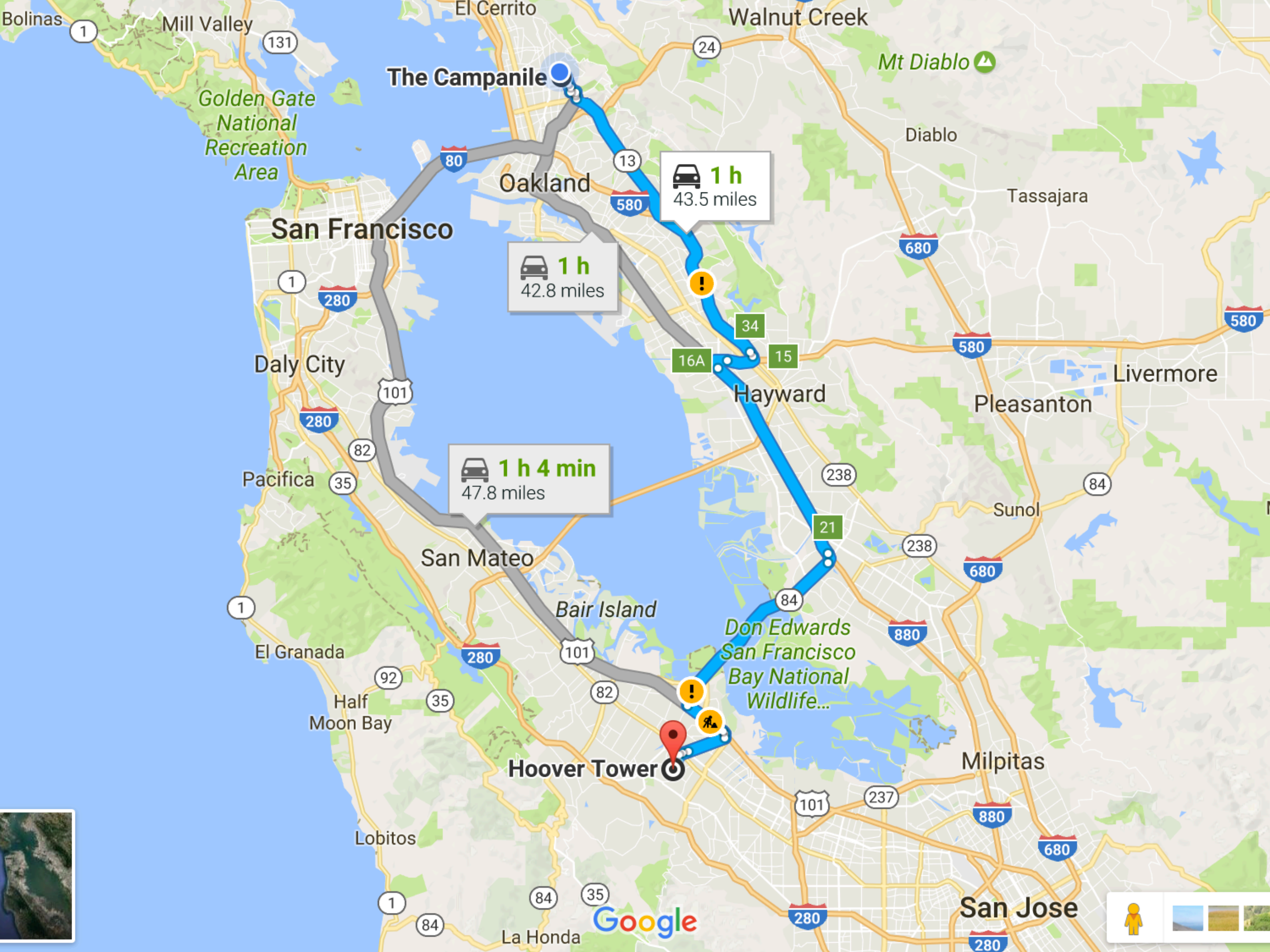Guerrilla section **Sat 12-3pm in 247 Cory** OOP + Inheritance + Orders of Growth

Homework 05 due Monday

- Homework party 6:30-8:30pm

Orders of Growth

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

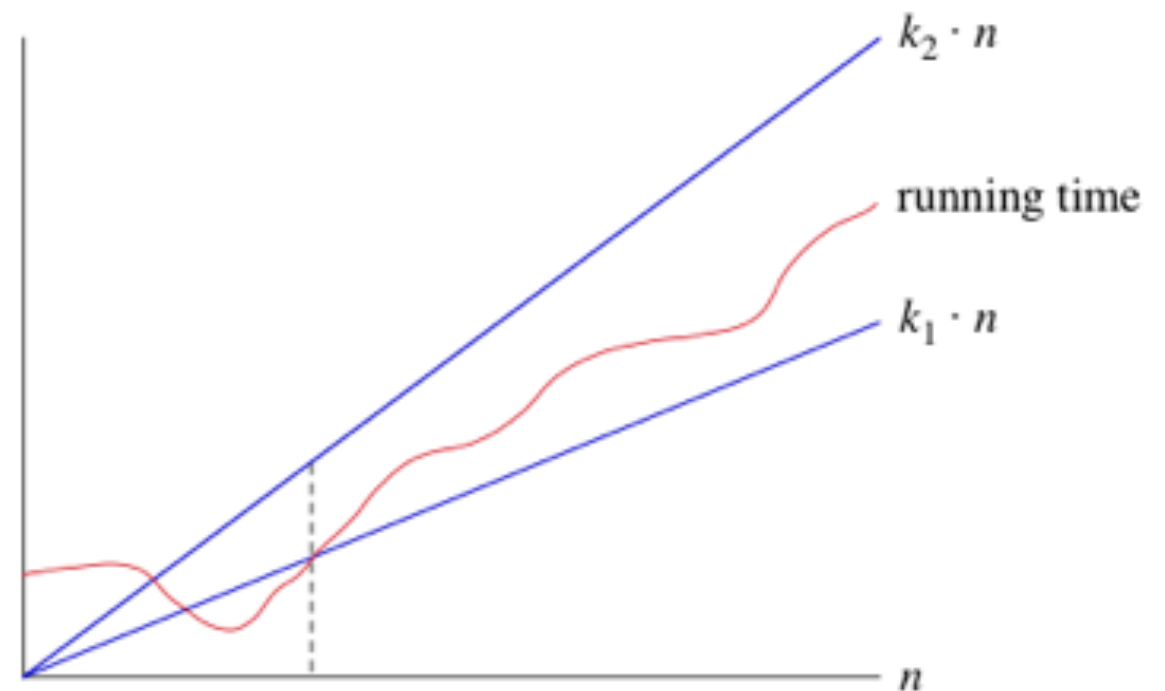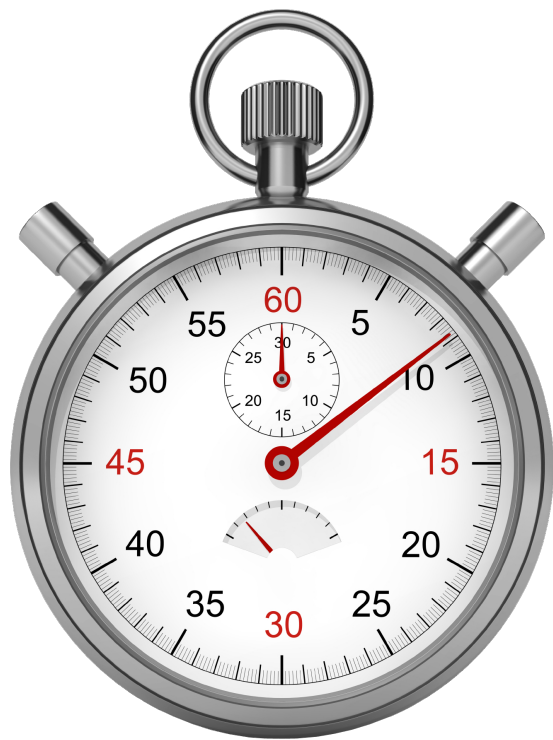Google's Deepmind AI beats Go world champion in first match
Engadget - 23 hours ago

More news for deepmind

Orders of Growth

How do we describe how fast a program is?

Orders of growth — "as input size changes, how does run time?"

Orders of Growth

Time (μsec) for problem size N	Max N Possible in			
	1 second	1 hour	1 month	1 century
$\lg N$	10^{300000}	$10^{1000000000}$	$10^{8 \cdot 10^{11}}$	$10^{9 \cdot 10^{14}}$
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

Simplify	Answer
$\theta(3n)$	$\theta(n)$ — ignore const factors
$\theta(n^3 + 1000n^2)$	$\theta(n^3)$ — larger term dominates
$\theta(\log n + n)$	$\theta(n)$ — larger term dominates
$\theta(n \log n + n)$	$\theta(n \log n)$ — larger term dominates

Just for "Fun"

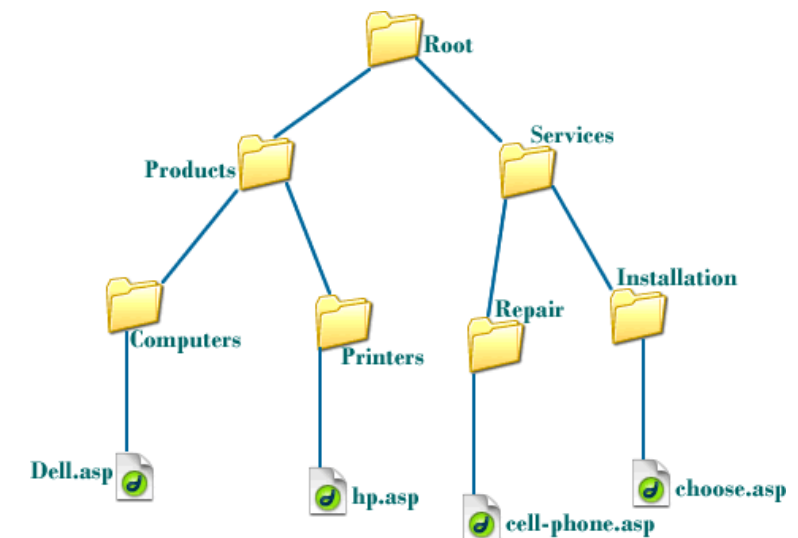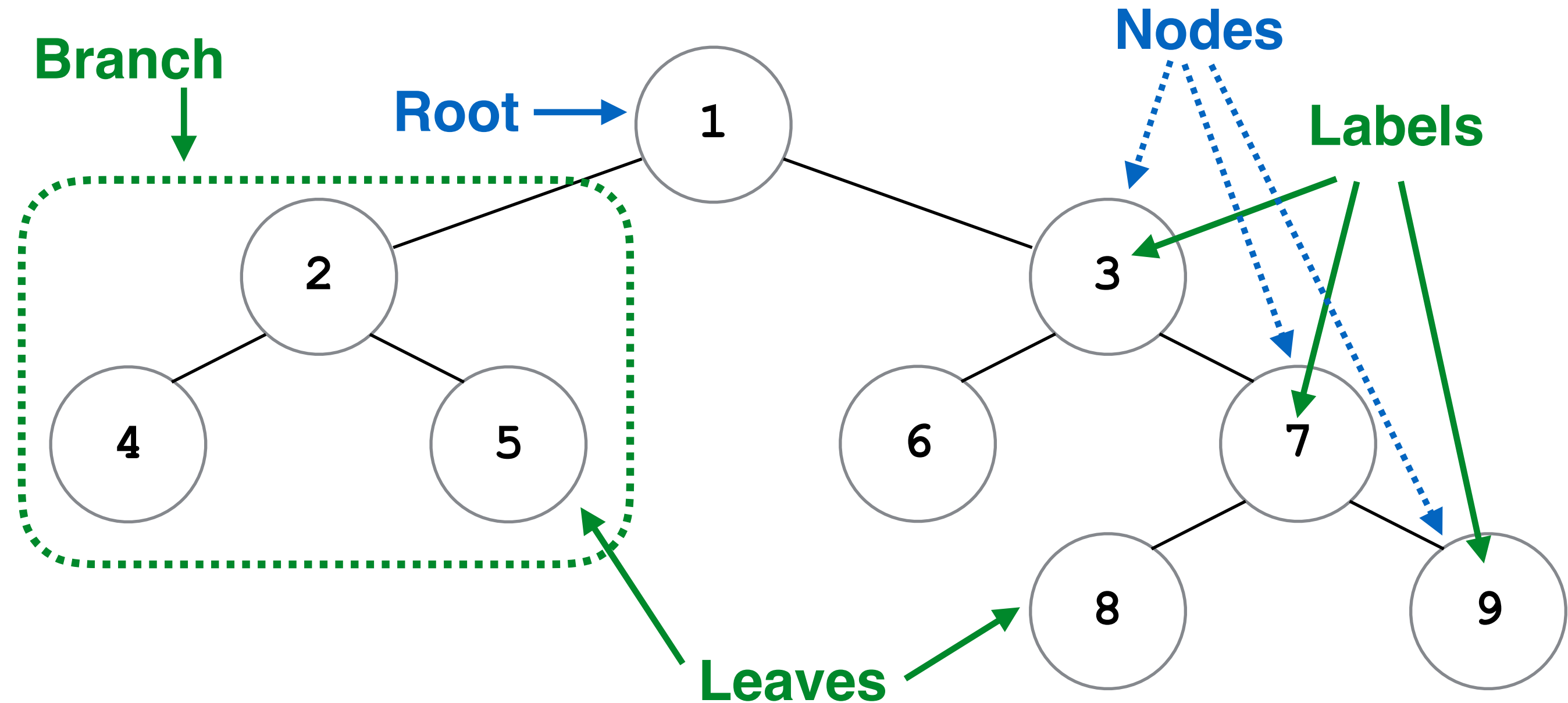| Question**	Answer
$\theta(\log_2 n) > \theta(\log_{10} n)$	**No** — Use change of base formula.
$\theta(n \log(n^8)) > \theta(n^2 \log(n^3))$	**No** — use log rules to get $\theta(n \log n)$ vs $\theta(n^2 \log n)$
$\theta(n \log n) < \theta((\log n)^{\log n})$	**Yes** — RHS is $n^{\log \log n}$. Or take log of both sides.

Trees

Trees

Storing things in order like a list is boring...

In real life, you **see trees everywhere!**

- Taking notes

- Directory structure on your computer

- Nature and stuff, I guess

Trees

Branch

Root → 1

Nodes

Labels

2　　　　3

4　　5　　6　　7

Leaves　　8　　　9

Credit: Based on Prof. DeNero's tree diagram [Fa 16 CS 61A]

Trees (OOP)

Constructor:

```
Tree(label, branches=[])
```

Selectors:

```
t.label, t.branches, t.is_leaf()
```

Trees (OOP)

Can now also modify labels (and branches)!