



# Welcome to 61A Lab!

---

We will begin at **5:10!**  
Slides: **[cs61a.bencuan.me](https://cs61a.bencuan.me)**

# Announcements

---

- Midterm next Thurs.
  - No discussion this week!
  - Have a nice spring break :)
- Homework 5 due today

# The Plan

---

- ▣ Midterm info and resources
- ▣ Efficiency minilecture
- ▣ Lab hints + work time
- ▣ Optional: conceptual review and past midterm walkthroughs
  - ▣ Trees/tree recursion especially
- ▣ Optional: discussion problem backlog

# Midterm info

---

# Midterm logistics

---

**Time:** Thursday, 3/17 8-10pm

**Bring:** 2 double-sided cheat sheets, pen(cil)s, SID, water bottle, mask

**If not feeling well:** email [cs61a@berkeley.edu](mailto:cs61a@berkeley.edu) ASAP to schedule alternate/online

# Midterm topics

---

## **From roughly most to least emphasized (not guaranteed to reflect actual midterm):**

1. Recursion and tree recursion
  - a. Partition problems (count coins, count stair ways)
  - b. Classic recursion problems (ping pong, hailstone)
2. Trees and Linked Lists
3. Lists and Mutability
  - a. Pop, append, extend, remove
  - b. What is and is not a mutation (disc7)
4. OOP (inheritance, design)
5. Iterators and Generators
  - a. Yield, yield from, next()
  - b. Map, filter, reduce
6. String representation (str, repr)
  - a. Probably only one small problem
7. Efficiency (linear, quadratic, exponential...)
  - a. Probably only one small problem

# Debugging problems

---

## **Somewhat new this semester!**

Go to <https://piazza.com/class/kxj8vcku7037li?cid=1588> for some practice problems

Make sure you're comfortable stepping through programs mentally / making environment diagrams!

# Resources

---

Also posted on [cs61a.bencuan.me](https://cs61a.bencuan.me)!

Ben's midterm studying strategy guide:

<https://cs61a.bencuan.me/Midterm-Tips-sp22-858964ddc43343cea52f6afbb2af05cf>

Tanay's list of useful midterms and problems:

<https://sparkling-swamp-b74.notion.site/CS-61A-Resource-Guide-6c4b98c53089424f9554fff9b1107698>

CSM study materials:

<https://docs.google.com/document/d/145kJIPtrbu410SYVhyznOV19gPCm5EOZ6XbtReOk33o/edit>





# Efficiency

---

go to notebook!  
(if you are viewing this slides  
yourself, go to [efficiency\\_notes.pdf](#))

# Lab Hints

---

- ▣ Iterators: call `next()` in a **while** loop
- ▣ Generators:
  - ▣ Iterative: **yield** in a **while** loop
  - ▣ Recursive: **yield from** a recursive call
  - ▣ Beware infinite generators! **do not** attempt to convert them into a list
- ▣ `isinstance(lnk, Link)` is useful for deep problems
- ▣ Go to previous slides for tree/link skeletons!

# Work Time!

---

[go.cs61a.org/ben-queue](https://go.cs61a.org/ben-queue)

