Sp18 CS 61B Discussion 3

Welcome!

Wayne Li

wli2@berkeley.edu

https://wayne-li2.github.io/

Announcements

- Project 1A due this Friday!
  - Come to OH! (Mine is Tuesday 1-3pm, SLC)

Before We Start...

- This section is an exam prep section! Plan accordingly.
  - Spend the next few weeks attending different discussions to find your favorite TA.
  - Please find a TA that suits your learning style! Don’t worry, we won’t be offended :)

“Quiz” Time!

Quiz Directions

- If you haven’t yet, please also **neatly** put your email address **outside the name box** if you want to be emailed!
- Bubble number **41**.

Aside

When to use a LinkedList?

When to use a LinkedList?

- Easy to add to front/back.
- Easy to remove from front/back.
- Hard to add/remove from middle (but hard for arrays!)
  - Can fix this with **hashing**
- Hard to search

When to really use a LinkedList?

- Practically, never. Why?

When to really use a LinkedList?

- In **61C**, you will learn about machine architecture, processors, and **caches**.
- LinkedLists have **really bad cache performance**.
- Source (it's in C++...): https://www.youtube.com/watch?v=YQs6IC-vgmo

Moral of the Story

- **Don’t forget about hardware**.

Exam Tips

Linked Lists (or Coding Problems in General)

- #1 Mistake: Start coding immediately.

Linked Lists (or Coding Problems in General)

- Example: Implement addFirst().
  - 1. Draw yourself an example.
  - 2. Draw what you want to happen.
  - 3. Convert your actions into code.

Linked Lists (or Coding Problems in General)

- #2 Mistake: Not Checking edge cases
  - Draw out common edge cases.
    - For example, small lists.

Onto Discussion

Q1: Flatten

- Visualizer: https://goo.gl/1sQ1fE

Q2: Skippify

- Visualizer: https://goo.gl/ZgoLcX

Q3: Remove Duplicates

- Visualizer: https://goo.gl/GFi37V