Sp18 CS 61B Discussion 11

Welcome!

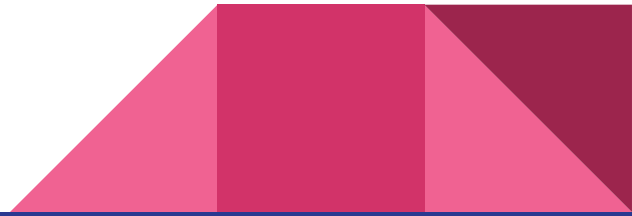
Wayne Li

wli2@berkeley.edu

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

Announcements

- Midterm II scores are out (@3262)
  - Regrade requests will open on **April 4th, at noon**
  - Regrade requests will close on **April 9th, at noon**
  - Special regrade instructions for FLIGHT
- Algorithm design problems (@3259)
- HW 4 (Puzzle Solver) due tomorrow 4/4
- Hope you all had a restful and fun spring break!



Quiz Instructions

- 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

Resilience

- If you didn’t get the score you wanted
  - **I’m here for you!**
- Want to meet with me specifically?
  - https://goo.gl/forms/3HyKxt0ZAWHKRmij2
  - Can chat about anything, serious or light!
- Want to meet with **any** TA (not me)? @3258

Feedback

- Not many responses :(
- New survey! (much shorter, 1 Q!)
- https://goo.gl/forms/qxZ8qzS47HB62wiE3
- Will give the option of non-anonymity.
- Give me criticism!! Nothing too small to bring up.

Feedback so Far

- Room’s too small!
  - Sorry… :(Trying to find a 61C TA that will switch rooms with me

Feedback so Far

- Speed: Just right -> Too Fast
  - Want to gather more data from this week!
  - Remember this is exam prep: come with the concepts learned from regular discussion!
  - If you’re struggling, email me on the side! More than happy to explain concepts.

Why is this Important?

Onto Discussion

Q1 (Warmup)

Runtimes

idea	addEdge(s, t)	for(w : adj(v))	printgraph()	hasEdge(s, t)	space used
adjacency matrix	$\Theta(1)$	$\Theta(V)$	$\Theta(V^2)$	$\Theta(1)$	$\Theta(V^2)$
list of edges	$\Theta(1)$	$\Theta(E)$	$\Theta(E)$	$\Theta(E)$	$\Theta(E)$
adjacency list	$\Theta(1)$	$\Theta(1)$ to $\Theta(V)$	$\Theta(V+E)$	$\Theta(\text{degree}(v))$	$\Theta(E+V)$

Key Idea: How to Represent Graphs

- Important real-life considerations.

Key Idea: How to Represent Graphs

- Adjacency Matrix
  - Fast **hasEdge(u, v)**. Probably the most common operation on a graph.
  - Ex: **Facebook - are A and B friends?**

Key Idea: How to Represent Graphs

- Adjacency Matrix
  - If graph is sparse, a lot of **wasted space**.
  - Ex: **Facebook - 2.2 billion users, most users have between 100-1000 friends**.

Question

- Should Facebook use an adjacency matrix?

Question

- Should Facebook use an adjacency matrix?
  - They pay someone a lot more than me to answer this question.

Key Idea: How to Represent Graphs

- Adjacency List
  - Slower **hasEdge(u, v)**.
  - But faster on everything else! Namely:
    - $O(V + E)$ instead of $O(V^2)$

Key Idea: How to Represent Graphs

- Adjacency List
  - Important: Know what **size of E** is relative to **size of V**.
  - Therefore, on dense graphs, **E ~ O(V^2)**, and adjacency lists lose their speed.

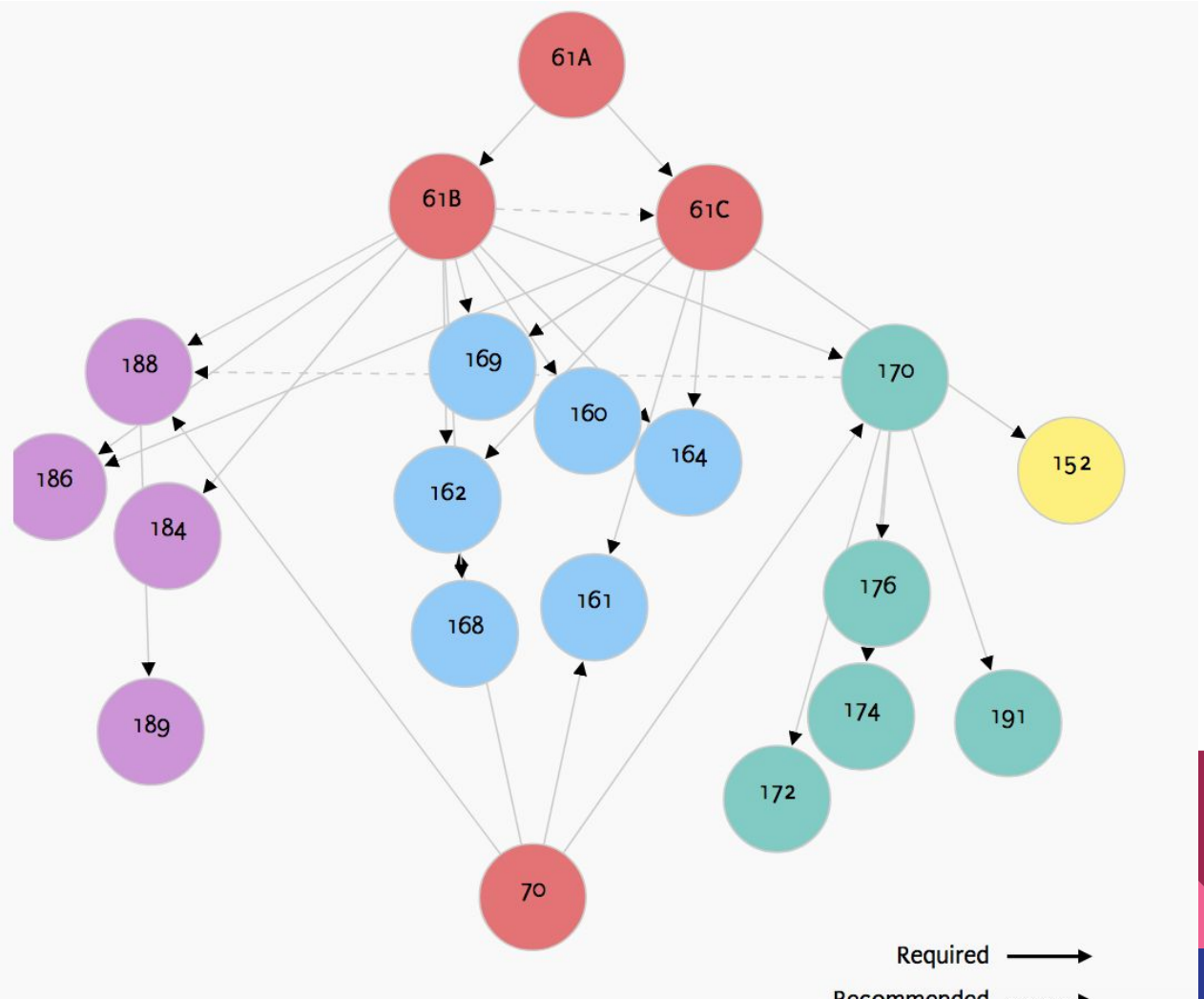
Key Idea: How to Represent Graphs

- Edge Set
  - Seems fast! Everything O(1) or O(E).
  - But **hasEdge(u, v)** is O(E).

Q2 (skip)

Q3

Berkeley CS Courses



Q4