Where to Go from Here
Outline for Today

● **Where We've Been**
  - What did we cover this quarter?

● **Your Questions**
  - Questions on any topics you'd like!

● **What Comes Next**
  - What's next in theory?

● **Final Thoughts**
  - Wrapping up the experience.
Where We've Been
Where We've Been

- Hybrid RMQ
- Fischer-Heun
- Aho-Corasick
- Suffix Trees
- Suffix Arrays
- B-Trees
- Red/Black Trees
- Augmented Trees
- Amortization
- Binomial Heaps
- Fibonacci Heaps
- Euler Tour Trees
- Splay Trees
- Count(-Min) Sketches
- Linear Probing
- Cuckoo Hashing
- vEB Trees
- \{x, y\}-Fast Tries
- Disjoint-Set Forests
- Dynamic Graphs
Key Theory Techniques from CS166

...with Relevant Life Advice.
Look for Easy Cases

- **Small problems are often easy to solve. Look for asymmetric divide-and-conquer strategies.**
  - Fischer-Heun RMQ
  - $x$-Fast and $y$-Fast Tries
  - The Prefix Parity Problem
  - Analysis of Disjoint-Set Forests
- This comes up in a ton of other contexts, especially if you look at more advanced graph algorithms or data structures.
Invest for the Future

- **Problems are rarely solved in isolation. Expect to do the same thing over and over.**
  - All of RMQ
  - Aho-Corasick Matchers
  - Suffix Trees
  - Suffix Arrays
  - Prefix Parity
- Many tough practical problems become easy if you can reuse the work you did in one context across a ton of other contexts.
Be Lazy

- *Sometimes it's good to let messes accumulate a bit. There's a reason we have dishwashers.*
  - Binomial heaps
  - Fibonacci heaps
  - Splay trees
  - Disjoint-Set Forests
  - Dynamic Graphs
  - \{x, y\}-Fast Tries
- It's good to plan for the future and to make sure that you don't get yourself into too big of a mess, though. 😊
Change Representation

- Don't overtrain on how the data "has" to be stored. Look for other approaches.
  - 2-3-4 Trees and Red/Black Trees
  - Disjoint-Set Forests
  - Suffix Arrays
  - Euler Tour Trees and Dynamic Graphs
  - Fibonacci Heaps.
- Train yourself to identify what's inherent to a problem and what's accidental. Often times, changing out the accidental component will make the solution present itself.
Teamwork Helps

- A single data structure might not be good enough, but a whole lot of them probably will be!
  - Count[-Min] Sketches
  - Tug-of-War Sketches
- See if you can amplify the power of a single estimate by getting a lot of them together.
Give Credit (and Blame) Fairly

- *Don't blame everyone for a failure, and give proper credit for success.*
  - Weight-Balanced Trees
  - Dynamic Graphs
- Teamwork is great – but be honest with yourself about it!
Find Independence

- *Events in the real world are correlated. Find ways to add small degrees of independence.*
  - Cuckoo Hashing
  - Linear Probing
- Remember that life experiences are not i.i.d. You will have a richer life if you do.
Enjoy Diversity of Opinion

- You can get to the same place in many different ways.
  - van Emde Boas Trees
  - y-Fast Tries
- Don't pigeonhole yourself into thinking there is “a” way to do something. Diversity of perspective is a good thing.
Your Questions
Where to Go From Here
More Theory Classes

- CS167
  - Transition to research-level algorithms.
- CS168
  - Survey of modern algorithmic techniques
- CS261 / CS361B
  - Advanced algorithm techniques.
- CS262
  - Algorithms and data structures for genomics.
- CS265
  - Randomized algorithms and data structures.
- CS267
  - Algorithms and data structures for graphs.
- CS362
  - Algorithms and data structures for modern data models.
Questions to Keep Asking

Can we change our perspective?
Can we change our constraints?
Can we do better?