

Your Next Week

Tuesday June 9

6:30 PM

- DUE Class 21 Lab
- DUE Class 21 Code Challenge
- DUE Class 22 Reading
- Class 22A

Wednesday June 10

6:30 PM

- Class 22B

MIDNIGHT

- DUE Class 22 Learning Journal

Thursday June 11

6:30 PM

- Co-working

Friday June 12

Saturday June 13

9:00 AM

- DUE Class 22 Lab
- DUE Class 22 Code Challenge
- DUE Class 23 Reading
- Class 23
- Interview Prep

MIDNIGHT

- DUE Class 23 Learning Journal

Sunday June 14

6:30 PM

- Co-working

Monday June 15

Tuesday June 16

9:00 AM

- DUE Class 22 Lab
- DUE Class 22 Code Challenge
- DUE Class 23 Reading
- Class 23
- Interview Prep

MIDNIGHT

- DUE Class 23 Learning Journal

What We've Covered

<p><i>Module 01</i> Javascript Fundamentals and Data Models</p> <p>C01 — <i>Node Ecosystem, TDD, CI/CD</i> C02 — <i>Classes, Inheritance, Functional Programming</i> C03 — <i>Data Modeling & NoSQL Databases</i> C04 — <i>Advanced Mongo/Mongoose</i> C05 — <i>DSA: Linked Lists</i></p>	<p><i>Module 02</i> API Servers</p> <p>C06 — <i>HTTP and REST</i> C07 — <i>Express</i> C08 — <i>Express Routing & Connected API</i> C09 — <i>API Server</i> C11 — <i>DSA: Stacks and Queues</i></p>	<p><i>Module 03</i> Auth/Auth</p> <p>C10 — <i>Authentication</i> C12 — <i>OAuth</i> C13 — <i>Bearer Authorization</i> C14 — <i>Access Control (ACL)</i> C15 — <i>DSA: Trees</i></p>	<p><i>Module 04</i> Realtime</p> <p>C16 — <i>Event Driven Applications</i> C17 — <i>TCP Server</i> C18 — <i>Socket.io</i> C19 — <i>Message Queues</i> C20 — <i>Midterms Prep</i></p> <p><i>Midterms</i></p>
<p><i>Module 05</i> React Basics</p> <p>C21 — <i>Component Based UI</i> C22 — React Testing and Deployment C23 — <i>Props and State</i> C24 — <i>Routing and Component Composition</i> C25 — <i>DSA: Sorting and HashTables</i></p>	<p><i>Module 06</i> Advanced React</p> <p>C26 — <i>Hooks API</i> C27 — <i>Custom Hooks</i> C28 — <i>Context API</i> C29 — <i>Application State with Redux</i> C30 — <i>DSA: Graphs</i></p>	<p><i>Module 07</i> Redux State Management</p> <p>C31 — <i>Combined Reducers</i> C32 — <i>Asynchronous Actions</i> C33 — <i>Additional Topics</i> C34 — <i>React Native</i> C35 — <i>DSA: Review</i></p>	<p><i>Module 08</i> UI Frameworks</p> <p>C36 — <i>Gatsby and Next</i> C37 — <i>JavaScript Frameworks</i> C38 — <i>Finals Prep</i></p> <p><i>Finals</i></p>

Code Challenge 21

Review

Sorting

- A very common coding need
- What happens when the size of the list (n) gets larger and larger?
- Various algorithms have different time complexities
- We'll cover the simplest one this class



Insertion Sort

- Imagine sorting playing cards in your hands
 - Work left to right, index 0 to $n-1$
 - For each index, move the value left until it doesn't need to go further

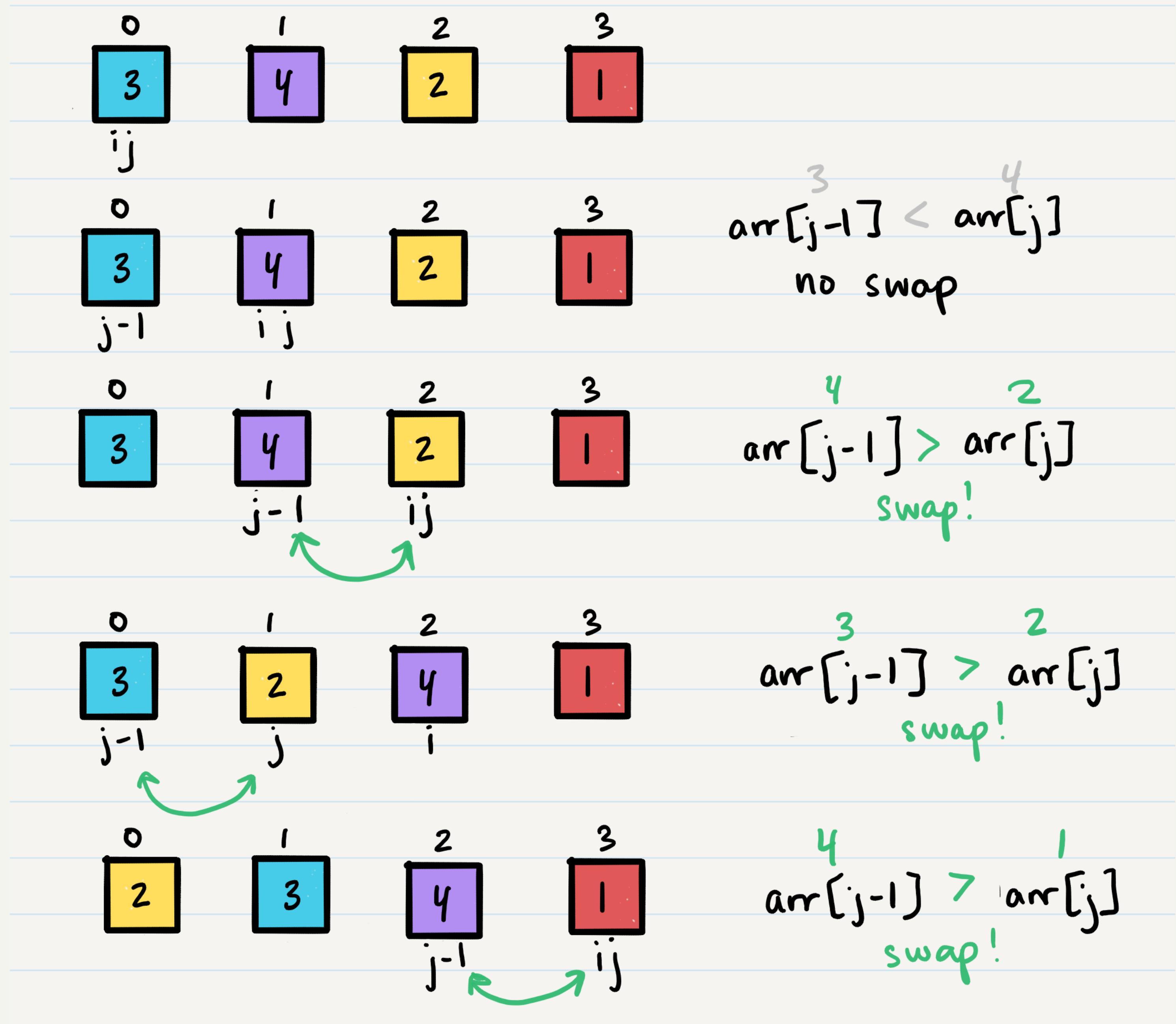


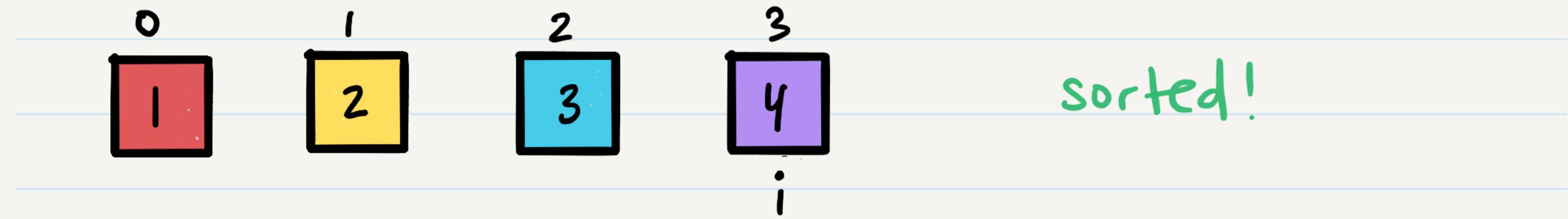
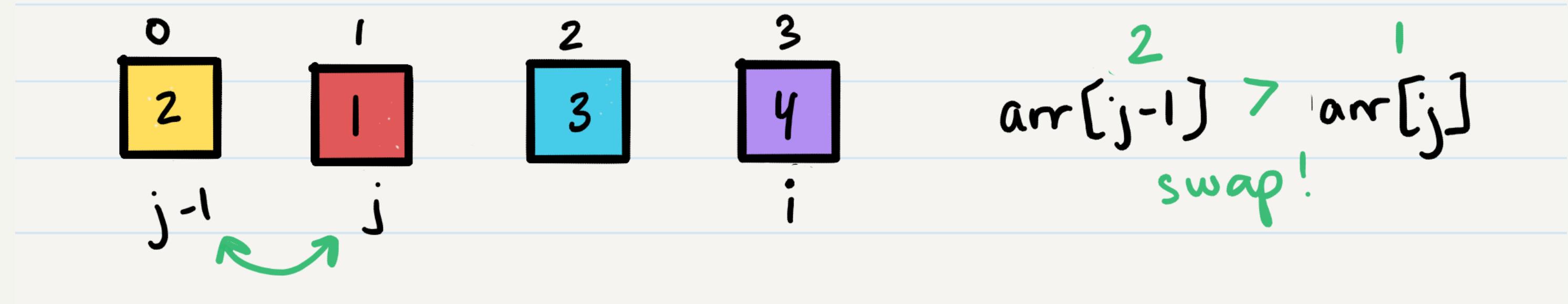
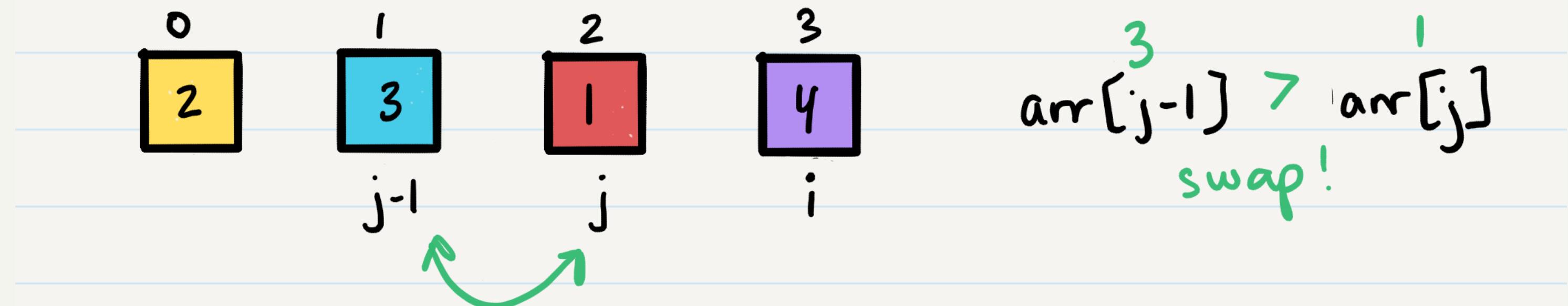
```
let i = 0;

while(i < arr.length) {
    let j = i;

    while(j > 0 && arr[j - 1] > arr[j]) {
        let temp = arr[j - 1];
        arr[j - 1] = arr[j];
        arr[j] = temp;
        j--;
    }

    i++;
}
```



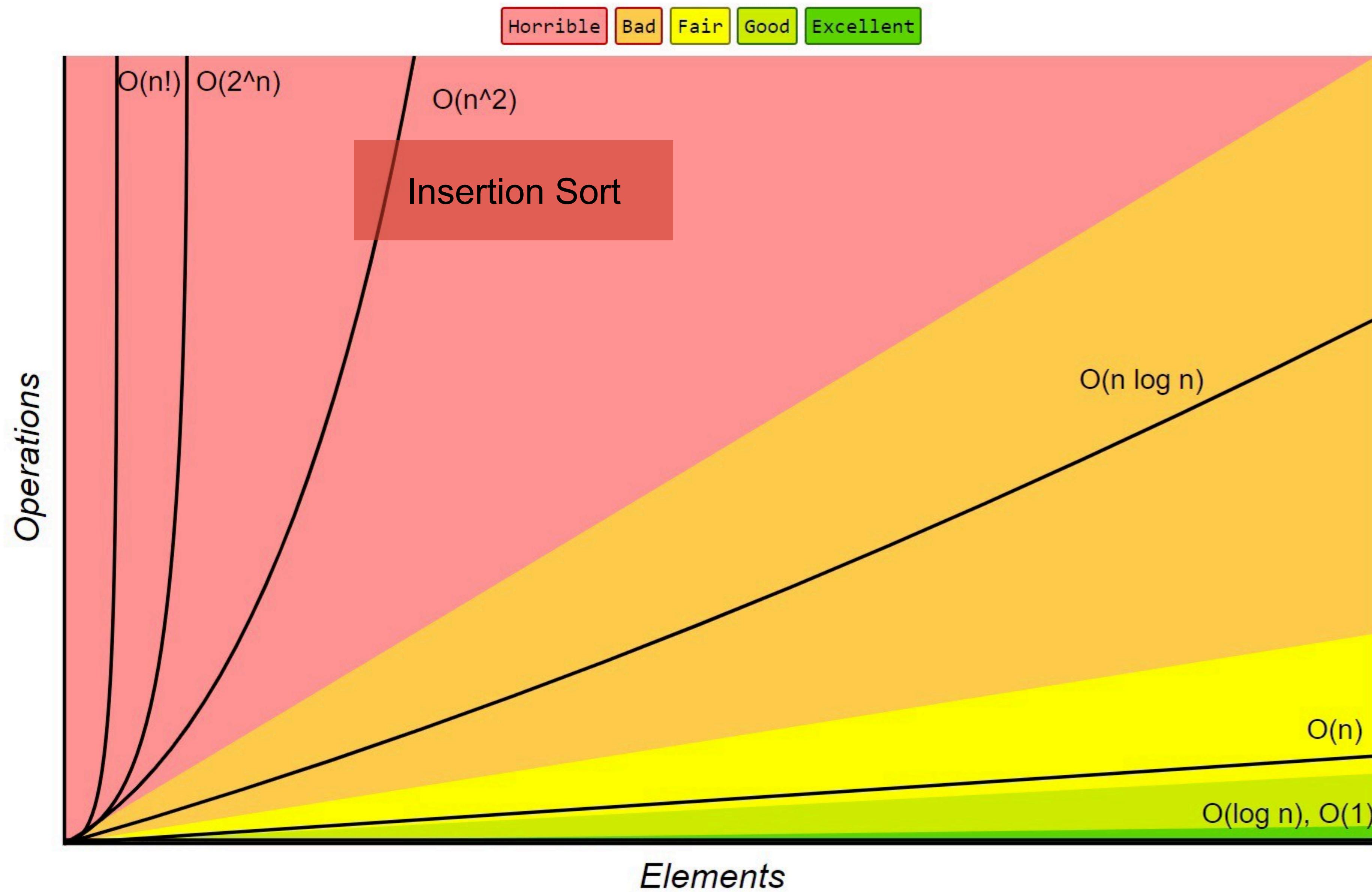


Insertion Sort

- Two while loops, so $O(n^2)$
- Not very efficient!
- Avoid if you can, though it's the simplest sort to implement



Big-O Complexity Chart



Lab 21 Review

Class 22

React Testing and Deployment

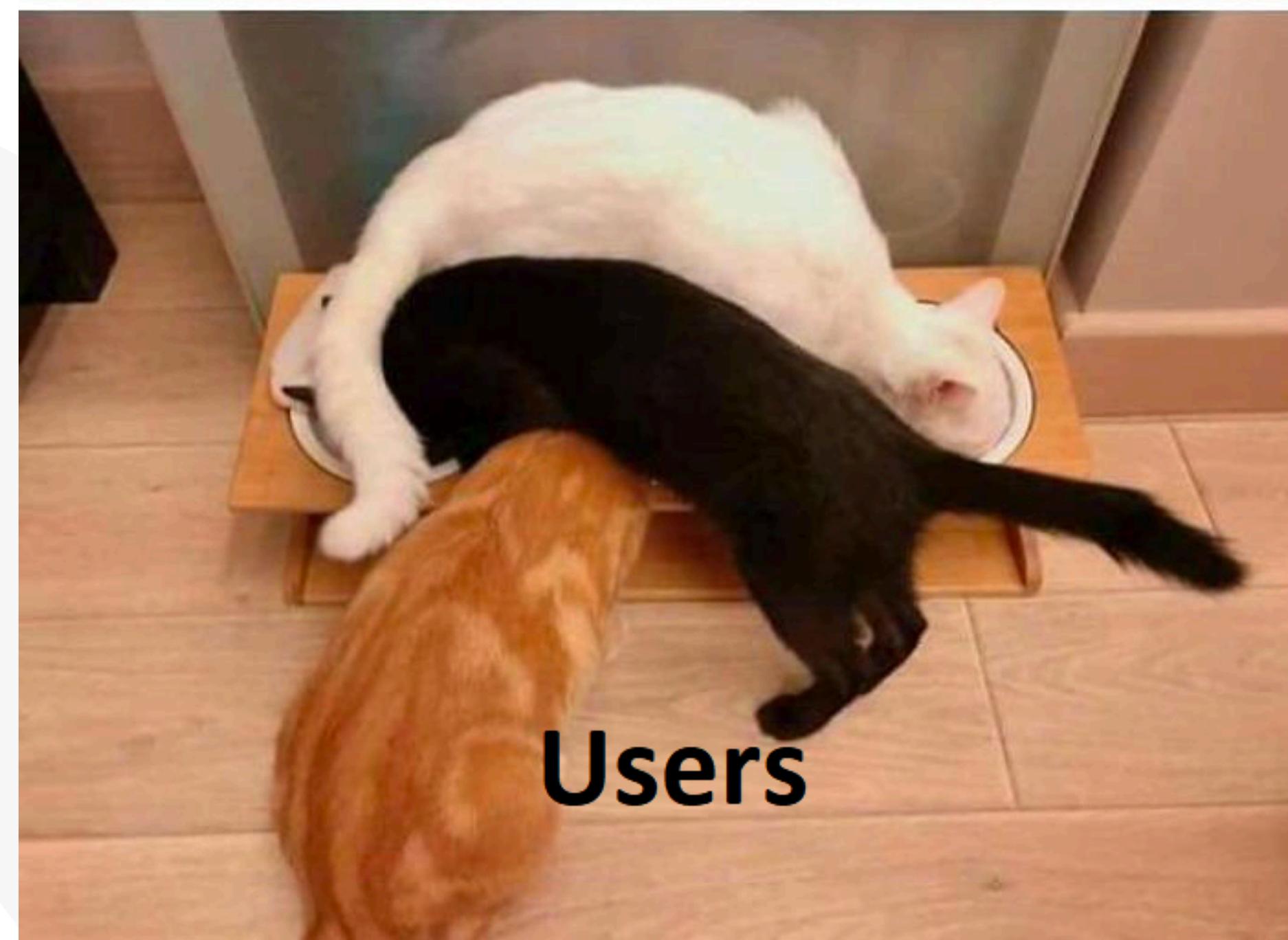
seattle-javascript-401n16

Testing UI

- We can check either
 - That the rendered HTML matches our expectations
 - Can be tricky with TDD
 - If the state and props match our expectations during user interaction



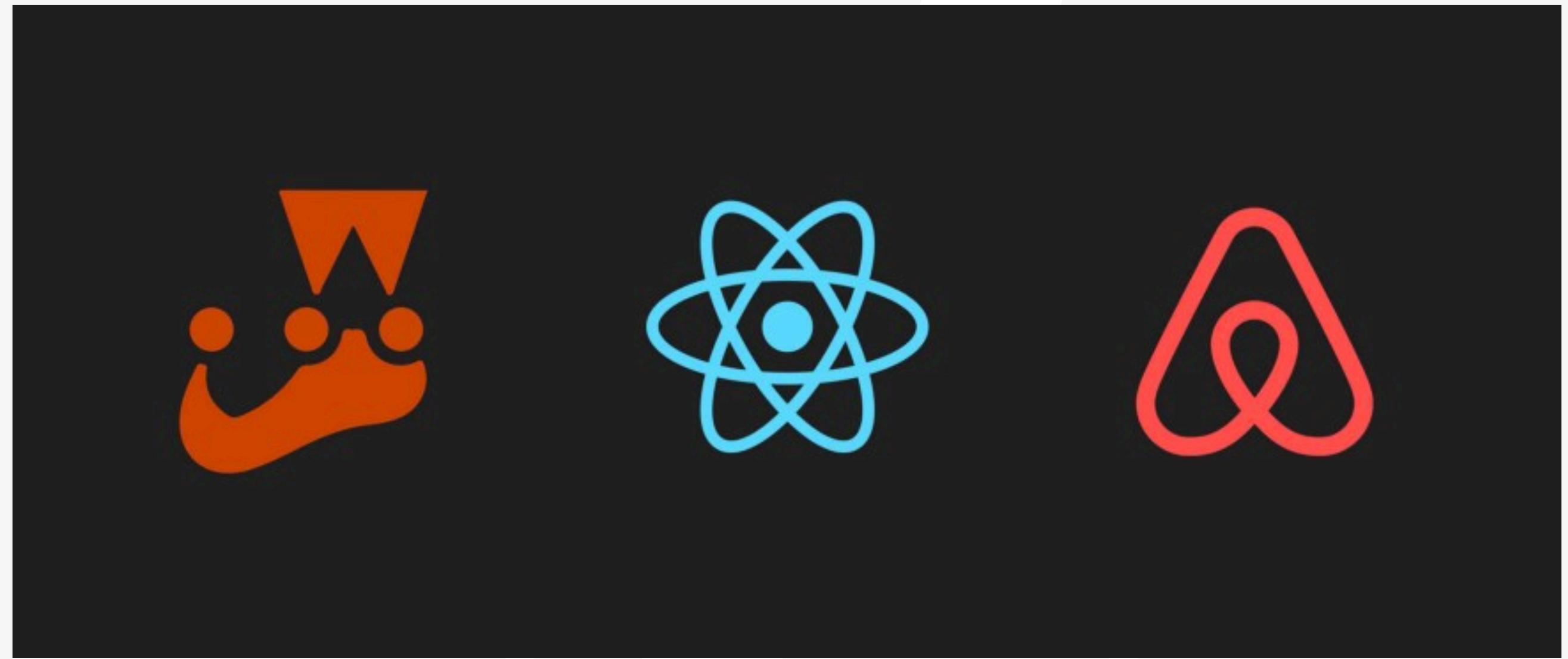
Developer: Makes a simple, intuitive UI



Users

Enzyme

- Another package we use in addition to Jest
- Allows us to search HTML using selectors (like JQuery!)
- We can fake user action such as button click or form input
- We can access component state



Lab 22 Overview