

Step 1: Alternative Tools Research

1. **Sentry** - an application monitoring platform that helps devs identify and fix production errors across every application in a company's stack
 - a. Value Adds - consolidates errors into larger issues categories for you. Shows the impact of specific problems, and even
 - b. Notable features - Reports show the exact commit that caused the error.
 - c. Drawbacks - pricing
 - d. Getting Started Instructions - Sentry offers a gorgeous and extensive set of resources for getting started using the service on 99 platforms. To use them simply navigate to their DOCS and then select your platform from a list. Documentation walks through installation, configuration, troubleshooting and more.
 - e. Notable Resources - there are numerous resources in the Sentry Library that serve as tutorials on how to use the service- video, audio, pdfs and demos are all included. They also have a sandbox that provides interactive demo material.
 - f. Maturity Evaluation - Sentry's earliest commit on GitHub looks like it was in 2012. And the company was founded in 2012. They are used by GitHub itself, Disney, doorDash, Reddit, Microsoft and Peloton to name a few.
2. **Jenkins**
 - a. Value Adds - it works for both CI & CD
 - b. Notable features - provides over 300 plugins to support building and testing in lots of environments. It's also open source.
 - c. Drawbacks - less robust discussion of logs and how to use them to fix code after its in production
 - d. Getting Started Instructions - they are relatively clear
 - e. Notable Resources - tutorials and documentation exist although they are not as user-friendly as Sentry's.
 - f. Maturity Evaluation - Jenkins io was also founded in 2012. I was able to find a list of companies that contribute to Jenkins. They are reportedly used by Netflix, Udemy, Facebook, Twitch, & Linked in. They call themselves "the leading open source automation server."

Step 2: Runtime Analysis

Array Size	doublerAppend	doublerInsert
tinyArray	90.074 µs	52.786 µs
smallArray	108.851 µs	50.281 µs
mediumArray	148.307 µs	150.762 µs
largeArray	530.391 µs	6.428509 ms

extraLargeArray	4.432608 ms	815.667022 ms
-----------------	-------------	---------------

Evaluation:

You can see from the chart above that the appending function starts off slower than the inserting function, but eventually, as array sizes grow it is able to append faster than it can insert. The appending function scales better than the inserting function.

Extra Credit Research: Both functions will be directly affected by the size of the array that is passed in. BUT unshift has to first alter where data is stored in an array and then push new data in. I think the runtime complexity of each of these functions is linear, but the inserting function takes longer because it requires more RAM as the size of the passed in array grows.