Algorithmic Problem Solving Conclusion

 (https://github.com/learn-co-curriculum/phase-1-algorithms-algorithmic-problem-solving-conclusion) (https://github.com/learn-co-curriculum/phase-1-algorithms-algorithmic-problem-solving-conclusion/issues/new)

Learning Goals

- Review what you learned in this section

Conclusion

Congrats on getting through this section! You've learned some extremely valuable skills in terms of being able to tackle challenging problems and identify what good code looks like. You're well on your way to mastering your first algorithm interview!

To recap, we covered these learning objectives:

- Define what "algorithm" means
- Develop a process for algorithmic problem solving
- Understand the usefulness of Big O notation
- Identify the Big O time and space complexity of a given algorithm
- Recognize common Big O runtimes: constant, logarithmic, linear, and quadratic

Where We're Going From Here

We'll revisit algorithms formally once again in Phase 4, where we'll cover another fundamental topic to help prepare for technical interviews: **data structures**. In the meantime, it's important that you continue practicing what you learned in this section and in particular, continue refining your problem solving process.

From now until we return to algorithms in Phase 4, your main priority should be learning the required material from each upcoming Phase and mastering the practical skills you'll need to become a software engineer. When you're working on labs, keep in mind what you learned in this section!

However, if you're able to make the time, it's also a great idea to dedicate a little time each day (say 30 to 60 minutes) to continuing with your algorithm studies.

Here are some resources to help:

Flatiron Data Structures and Algorithms Canvas Course

Make sure to check out the Data Structures and Algorithms course on Canvas. Ask your instructor for access if you don't already have it. This course was designed to give students short problems to work on throughout their time at Flatiron. There's also a version of this course you'll get access to upon graduation, so don't worry if you aren't able to complete everything during the program.

Our recommendation is to work on the following sections between now and Phase 4:

- Week 1: Starter Algorithms
- Week 2: Recursion
- Week 3: Additional Practice
- Week 4: Big O Notation
- Week 5: Sorting Algorithms

The material from Week 6 and onward gets into Data Structures, which we'll cover in more depth in Phase 4.

Work On Practice Exercises

Here are some good practice exercises with walkthrough videos from Flatiron instructors:

- Matched Substrings:
    - **Code** ⤷ **(https://leetcode.com/playground/SeqrcHRz)**
    - **Solution Video** ⤷ **(https://www.youtube.com/watch?v=s7u-0K4SxUA)**

▷

[(https://www.youtube.com/watch?v=s7u-0K4SxUA)](https://www.youtube.com/watch?v=s7u-0K4SxUA)

- Two Sum:
  - **Code ⤷ [(https://leetcode.com/problems/two-sum)](https://leetcode.com/problems/two-sum)**
  - **Solution Video ⤷ [(https://www.youtube.com/watch?v=qDEkdD4MfCY)](https://www.youtube.com/watch?v=qDEkdD4MfCY)**

    ▷

    [(https://www.youtube.com/watch?v=qDEkdD4MfCY)](https://www.youtube.com/watch?v=qDEkdD4MfCY)

- Reverse Integer:
  - **Code ⤷ [(https://leetcode.com/problems/reverse-integer/)](https://leetcode.com/problems/reverse-integer/)**
  - **Solution Video ⤷ [(https://www.youtube.com/watch?v=P_98V9dwWHc)](https://www.youtube.com/watch?v=P_98V9dwWHc)**

    ▷

    [(https://www.youtube.com/watch?v=P_98V9dwWHc)](https://www.youtube.com/watch?v=P_98V9dwWHc)

- Valid Palindrome:
  - **Code ⤷ [(https://leetcode.com/problems/valid-palindrome/)](https://leetcode.com/problems/valid-palindrome/)**
  - **Solution ⤷ [(https://www.youtube.com/watch?v=db7bp1LGkfU)](https://www.youtube.com/watch?v=db7bp1LGkfU)**

    ▷

    [(https://www.youtube.com/watch?v=db7bp1LGkfU)](https://www.youtube.com/watch?v=db7bp1LGkfU)

- Old-School Text Message Keystroke Counter Algorithm
  - **Code ⤷ [(https://www.codewars.com/kata/multi-tap-keypad-text-entry-on-an-old-mobile-phone)](https://www.codewars.com/kata/multi-tap-keypad-text-entry-on-an-old-mobile-phone)**
  - **Solution ⤷ [(https://www.youtube.com/watch?v=g2a03A_ORzs)](https://www.youtube.com/watch?v=g2a03A_ORzs)**

▷

[(https://www.youtube.com/watch?v=g2a03A_ORzs)](https://www.youtube.com/watch?v=g2a03A_ORzs)

For additional practice, here are some more resources:

- **Leetcode** ⬑ **[(https://leetcode.com/)](https://leetcode.com/)**
  - **Curated list of 100 problems for Flatiron students** ⬑ **[(https://leetcode.com/list/5r99deem/)](https://leetcode.com/list/5r99deem/)**
- **HackerRank** ⬑ **[(https://www.hackerrank.com/)](https://www.hackerrank.com/)**
  - **10 Days of JavaScript** ⬑ **[(https://www.hackerrank.com/domains/tutorials/10-days-of-javascript)](https://www.hackerrank.com/domains/tutorials/10-days-of-javascript)**
  - **Interview Preparation Kit** ⬑ **[(https://www.hackerrank.com/interview/interview-preparation-kit)](https://www.hackerrank.com/interview/interview-preparation-kit)**
- **codewars** ⬑ **[(https://www.codewars.com/)](https://www.codewars.com/)**