

# CMPS 6610 Lab 01

In this recitation, we will investigate asymptotic complexity.

Some prompts will require you to edit `main.py` and others will require answers will go in `answers.md`.

Refer back to the README.md for instruction on git, how to test your code, and how to submit properly to get all the points you've earned.

## Comparing search algorithms

In this lab, we'll compare the running times of `linear_search` and `binary_search` empirically.

1. In `main.py`, the implementation of `linear_search` is already complete. Your task is to implement `binary_search`. Implement a recursive solution using the helper function `_binary_search`.
2. Test that your function is correct by calling from the command-line `pytest main.py::test_binary_search`
3. Write at least two additional test cases in `test_binary_search` and confirm they pass.
4. Describe the worst case input value of `key` for `linear_search`? for `binary_search`?

**Enter your answer in answers.md**

5. Describe the best case input value of `key` for `linear_search`? for `binary_search`?

**Enter your answer in answers.md**

6. Complete the `time_search` function to compute the running time of a search function. Note that this is an example of a “higher order” function, since one of its parameters is another function.
7. Complete the `compare_search` function to compare the running times of linear search and binary search. Confirm the implementation by running `pytest main.py::test_compare_search`, which contains some simple checks.
8. Call `print_results(compare_search())` and paste the results here:

**Enter your answer in answers.md**

9. The theoretical worst-case running time of linear search is  $O(n)$  and binary search is  $O(\log_2 n)$ . Do these theoretical running times match your empirical results? Why or why not?

**Enter your answer in answers.md**

10. Binary search assumes the input list is already sorted. Assume it takes  $O(n \log n)$  time to sort a list of length  $n$ . Suppose you know ahead of time that you will search the same list  $k$  times.

- What is worst-case complexity of searching a list of  $n$  elements  $k$  times using linear search?

**Enter your answer in answers.md**

- For binary search?

**Enter your answer in answers.md**

- For what values of  $k$  is it more efficient to first sort and then use binary search versus just using linear search without sorting?

**Enter your answer in answers.md**