

# Lab # 2 Empirical Testing

CISC 233 Essential Algorithms

## Task Description

Write a program to run a series of tests to illustrate the big-O efficiency of *linear search* and *binary search* in an array. You can write your own code for linear search and binary search, or you can find code online, as long as you credit the source. Your code should be commented and should clearly indicate any parts of the code that were taken from other sources. At a minimum comments should include:

- Header comments describing purpose of code and include your name as developer
- comments to describe the function of any code that is not intuitively obvious
- comments to indicate source of any code that you've gotten from other sources

The basic structure of the program:

- Repeat the following, starting with smaller arrays, increasing size by a constant amount in subsequent trials:
  - Create and fill an array with random data with appropriate characteristics (items ordered for the case of binary search, unordered for linear search)
  - Search for some particular item in the array, measuring the time it takes to do so in milliseconds
- Output of the program should be minimal in the following form (which can be copied into a CSV file for further analysis):
  - First line of output has column headings `array_size` and `search_time`, separated by a comma.
  - Each subsequent line has two integer values indicating the `array_size` and `search_time` for tests.

## Considerations

- array sizes to be tested
- range of values in array; duplicates allowed? percent unique
- range of values searched for
- algorithms to use

After you have the basic structure working, implement the following

- Instead of just running one trial at each size, run number of trials at each size and report the average search\_time. Use a variable num\_trials to indicate how many trials will be run at each array size.
- Use variables min\_size, max\_size, and step\_size to control the minimum and maximum array sizes to be tested along with the step sizes.
- As a bonus, have the output go directly to a csv file with an informative name, including the date and time of the trial run. Such a file can then be opened directly by Excel for further analysis.

## Submission

- A single python source file with all necessary code named yourlastname\_searchtests.py
- A lab report with the following sections:
  - title page with class number and title, lab number and title, your name and date
  - overview section summarizing the lab task
  - results section showing the results from informative runs illustrating the difference in efficiency between linear and binary search.
    - this section should include graphs showing the different growth rates, along with the parameters used to produce the graphs. (the graphs themselves need not be produced by Python – you can take the output and import it into Excel or Google Sheets to create the graphs).
  - discussion section describing the results in terms of Big-O of the two algorithms