Abstract

With the goal of further understanding arrays in Python compared to those of other languages, this experiment will focus on the performance of the sorting algorithm binary search with varying input sizes. The test has some inconsistencies for two input sizes, however most of the trial runs followed the expected pattern of increasing the runtime as the input sizes increased. Compared to the compiled languages python took a much longer time in order to run binary search upon its one-dimensional array since it is an interpreted language.

Experiments

*Note: for easy comparison the same tests were done in languages: Python, Java, JavaScript, and C++

- Package used
    - `time` package used in order to measure how long the specific test took to run in nanoseconds
- One dimensional array
    - Array input sizes:
        - 1,875,000
        - 3,750,000
        - 7,500,000
        - 15,000,000
        - 30,000,000
    - Binary Search value
        - 500
    - Array content
        - Contains only index+1
            - Arr[0] = 1
            - Arr[127] = 128 ...etc.

A one dimensional array will be created and populated based on the input size given. The time it took for each array to complete the binary search on the value 500 is recorded each time as data. The time spent waiting mostly is due to the creation of an array with the specified size and data. Ran the code using an online compiler (https://www.onlinegdb.com/online_python_compiler) since I was getting results of 0 second runtimes within the PyCharm ide.

Results

```
Average of 10 runs for one-dimensional array of size 30000000 : 4.4679641723632814e-05 seconds
Average of 10 runs for one-dimensional array of size 15000000 : 3.495216369628906e-05 seconds
Average of 10 runs for one-dimensional array of size 7500000 : 3.528594970703125e-05 seconds
Average of 10 runs for one-dimensional array of size 3750000 : 3.533363342285156e-05 seconds
Average of 10 runs for one-dimensional array of size 1875000 : 2.696514129638671 7e-05 seconds


...Program finished with exit code 0
Press ENTER to exit console.
```

Array input size	Time (seconds)
1,875,000	$2.7 * 10^{-5}$
3,750,000	$3.53 * 10^{-5}$
7,500,000	$3.53 * 10^{-5}$
15,000,000	$3.5 * 10^{-5}$
30,000,000	$4.47 * 10^{-5}$

With the exception of the trial run for 15 million and 7.5 million, the time increases as the input size increases as expected.

Conclusion

Given the results of the trials if the test was run more than ten times per input size a more accurate answer could have been achieved, and the times should follow the expected pattern of increasing with the input size. Compared to the other two compiled languages it took python a substantially larger time in order to run binary search on the input sizes, however this was to be expected. Since python is an interpreted language unlike that of a compiled language, it is executed line by line thus contributing to the slow runtimes that were received as data.