

Test Plan: Search Coding Challenge

1. Introduction

This test plan is for the AeroFS program challenge. The challenge entailed writing a `Search()` function that performs linear search of a sorted array based on a given set of criteria.

2. Features To Be Tested

- *Correctness against requirement specifications*
- *Performance in terms of CPU time with respect to input array size*

3. Assumptions

- *The input is a non-empty sorted array of integers with no duplicate values*

4. Approach

I implemented the `Search()` function in Java. JUnit serves as my testing framework. I run a set of functional tests (`StaticTests.java` in the `test` package) using the given test cases. These static test cases check for basic functionality, but they are by no means exhaustive.

To test the functionality more completely, I created `DynamicTest.java` (also in the `test` package). This test generates various-sized, sorted arrays with random intervals between each element. It then tests all possible combinations of the search criteria to ensure correctness.

My last goal was to test performance, which I did using the CPU time capabilities of the `java.lang.management` package. I integrated this into `DynamicTest.java` to get an average of the CPU time elapsed during each iteration testing the `Search()` function with a certain array size.

Sample output is shown on the following page.

Sample Output:

Average (10 iterations) cpu time spent searching each 10-element array:1.9E-6 seconds.

Average (10 iterations) cpu time spent searching each 100-element array:2.1E-6 seconds.

Average (10 iterations) cpu time spent searching each 1000-element array:4.8E-6 seconds.

Average (10 iterations) cpu time spent searching each 10000-element array:3.5E-5 seconds.

Average (10 iterations) cpu time spent searching each 100000-element array:3.384E-4 seconds.

Average (10 iterations) cpu time spent searching each 1000000-element array:0.0032302 seconds.

5. Conclusion

The functional testing confirmed that the Search() function's correctness, and the CPU time measurements confirm that the Search() function performs well enough for most applications.

However, the CPU time measurements used are not an accurate enough tool for production-level testing. They are better suited in development for showing the effectiveness of the Search() function's algorithm with different input sizes.