

MModal Take Home Assignment

Table of Contents

- [Running and Compiling](#)
- [Assumptions](#)
- [Space and Time Complexity Analysis](#)

Running and Compiling

To run the program please navigate to the folder containing Compute.java and Compute.class and run (Example) ->

```
java Compute 3 0 1 2 3 4
```

This example was taken from the problem description.

If you need to recompile please run ->

```
javac Compute.java
```

Assumptions

- This program assumes the input variables are input as number separated by spaces. It has a certain amount of error catching, but **the input does not match the exact syntax of the problem description**, I wasn't sure how important that was.
- The program should provide double point precision and the output is formatted based off of the project description.

Space and Time Complexity Analysis

Space Analysis

In the for loops I had a choice whether or not to optimize for performance or to optimize for memory used. Doing more operations on the same variable all at once slowed performance (marginally) but benefitted space usage because there weren't separate temporary variables being created.

Because calculating the average for an input requires the previous average calculations, each additional input value may double the processing time if the previous averages are not cached to refer to directly.

Without caching the previous averages The Big-O notation would be:

```
O(2N).
```

With caching of the previously calculated values the Big-O notation would be:

```
O(N2). I think :)
```