Program Structures and Algorithms

Spring 2023(SEC 03)

NAME: Vipul Rajderkar

NUID: 002700991

**Task:**

Part-1: Implement (repeat, getClock, and toMillisecs) methods in the Timer class

Part-2: Implement InsertionSort in InsertionSort class

Part-3: Implement the main method that can calculate execution time for random, ordered, partial ordered, and reverse-ordered arrays.

**Relationship Conclusion:**

After benchmarking insertion sort on 4 different types of arrays following relationship was concluded basis execution time:

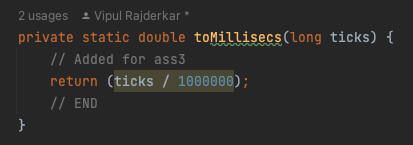
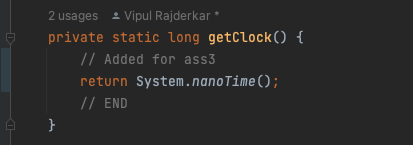
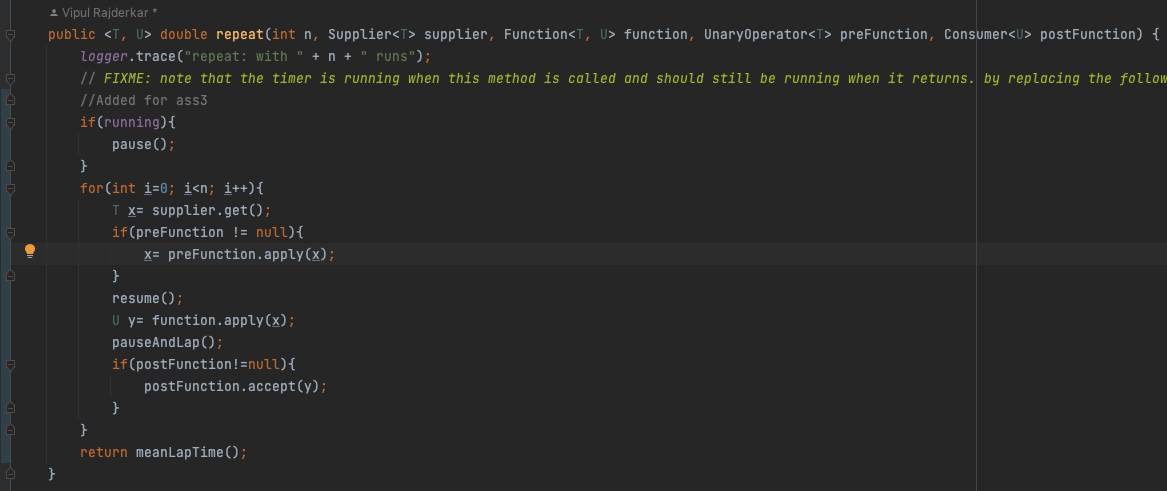
Ordered < Partially Ordered < Randomly Ordered < Reverse Ordered

(\*For a smaller number of inputs, the algorithm performs better with a partially sorted array than with a randomly sorted array. Please refer to the graphical representation in a later section)

**Evidence to support that conclusion:**

Part 1: Timer class methods

Code Snippet:



Graphical user interface, text

Description automatically generated

Part 2: Implement Insertion Sort

Code Snippet:

Text

Description automatically generated

Part 3: Main method to get execution time with different inputs

Text

Description automatically generated

**Graphical Representation:**

Timing Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **Random** | **Ordered** | **Partially Ordered** | **Reverse Ordered** |
| 1000 | 3.76 | 0.42 | 0.3 | 0.4 |
| 2000 | 2.02 | 0.36 | 0.26 | 0.44 |
| 4000 | 1.44 | 0.3 | 0.52 | 1.1 |
| 8000 | 3.32 | 0.4 | 1.28 | 3.78 |
| 16000 | 5.98 | 0.4 | 7.02 | 15.52 |

**Unit Test Screenshots:**

**BechmarkTest:**

**Graphical user interface, text

Description automatically generated**

**TimerTest:**

**Text

Description automatically generated**

**InsertionSortTest:**

Text

Description automatically generated