Merged Considerations for Sorting Algorithm Performance Testing

October 15, 2024

1 Purpose

Decide which sorting algorithm implementations to include in a library based on performance.

2 Questions

Which implementations run faster, and for what kind of inputs?

3 Dependent Variables

- Execution Time
- RAM Usage
- Number of Comparisons and Swaps

4 Independent Variables

- **Sorting Algorithm**:
  - BubbleSortUntilNoChange
  - BubbleSortWhileNeeded
  - QuickSortGPT
  - SelectionSortGPT
- **Array/Data Type**:
  - Integer, String, Float, Double
- **Array/Data Distribution**:
  - Random array

  - Pre-sorted array
  - Inverse sorted array
  - Array with repeated values
  - Nearly sorted arrays
- **Array Size**:
  - Variable sizes, randomly selected within a range for each experiment
- **Input Size**:
  - Length of numbers or strings (long or short strings)

5 Control Variables

- **Hardware Components**:
  - CPU, RAM, OS, storage, Java version, IDE
- **System Performance**:
  - CPU load, background processes, memory availability
- **Cache Utilization**:
  - Algorithms benefiting from cache performance may exhibit differences

6 Confounding Factors

- Hardware Differences (e.g., CPU speed, RAM)
- System Performance Variability (CPU load, memory usage)
- Measurement Techniques (e.g., `System.nanoTime()` vs. `System.currentTimeMillis()`)
- Java Versions (different JDK versions)

7 Randomization

- **Array Content**: Randomizing array contents with varying values and distributions.
- **Array Sizes**: Randomly selecting array sizes within a predefined range.
- **Pivot Selection in QuickSort**: Randomizing pivot selection to avoid worst-case scenarios.

8 Hypotheses

- **BubbleSortUntilNoChange**: Simple and efficient for small or sorted arrays but performs more comparisons overall.
- **BubbleSortWhileNeeded**: Reduces comparisons for already sorted or nearly sorted arrays.
- **QuickSortGPT**: Pivot selection influences performance (last element as pivot may cause worst-case performance for pre-sorted arrays).
- **SelectionSortGPT**: Consistent but inefficient for large datasets.

9 Other Considerations

- **Machine Warm-Up**: Perform warm-up iterations to stabilize system performance before recording results.
- **Mean vs. Median for Averages**:
  - Use mean if data is normally distributed.
  - Use median if data is skewed (e.g., due to random generation of arrays).
- **Graphical Representation**: Use graphs to visually compare results.