Università della Svizzera italiana | Institute of Computing CI

Experimentation and Evaluation **2024**

Students: Davide Frova, Costanza Rodriguez Gavazzi

Project 1

Due date: Monday, 11 November 2024, 11:59 PM

1. Abstract

//TODO
Short (120-130 words) summary of your entire report. Give the reader a quick idea of what you did and what the main findings were (if you prepare this report ahead of time, leave out the findings until after you finish the analysis).

2. Introduction

//TODO
Introduce the topic of investigation to the reader and motivate why you did the experiment. Note that in our case, writing "because I was told to by the course instructor" is not a valid answer. Please assume that you are trying to answer a certain relevant question and motivate its relevance. (In a "real" study report, you would need to also discuss any relevant prior research results here. Given our setting, however, we skip any "related work" consideration.) Your final paragraph of the introduction should outline your proposed experiment.

1. Hypotheses

1. Hypothesis 1: *The level of sortedness of the input data impacts the running time of the sorting algorithm.* The **independent variable** is the level of sortedness of the input data, which can vary between random, reversed, first-half-sorted, and last-half-sorted configurations. The **dependent variable** is the running time of the sorting algorithm. The **confounding variables** we identified are: the size of the dataset and the data type of its elements.

2. Hypothesis 2: *The size of the dataset impacts the running time of the sorting algorithm.* The **independent variable** is the size of the dataset, which can vary between 100, 1 000 and 10 000 elements. The **dependent variable** is the running time of the sorting algorithm. The **confounding variables** we identified are: the level of sortedness of the dataset and the data type of its elements.

3. Hypothesis 3: *The data type of the elements in the dataset impacts the running time of the sorting algorithm.* The **independent variable** is the data type of the elements in the dataset, which can vary between Int (4B), Long (8B), Float (4B), and Double (8B). The **dependent variable** is the running time of the sorting algorithm. The **confounding variables** we identified are: the level of sortedness of the dataset and the size of the dataset.

3. Method

1. Variables

- **Independent Variables**:
  - Level of sortedness of the input data: random, reversed, first-half-sorted, last-half-sorted.
  - Size of the dataset: 100, 1 000, 10 000 elements.
  - Data type of the elements in the dataset: Int (4B), Long (8B), Float (4B), Double (8B).
- **Dependent Variables**: Running time of the sorting algorithm.
- **Control Variables**:
  - **System**: The experiment was conducted on a MacBook Air with chip M1, 8GB of RAM and MacOS Sequoia 15.1.
  - **Programming Language**: The experiment was conducted using OpenJDK 21.0.4.
  - **IDE**: The experiment was conducted using VSCode 1.92.1.
  - **Running Processes**: The experiment was conducted with no other user processes running in the background.
  - **Code**: The experiment was conducted using the same code for all the combinations of variables.

2. Design

- **Type of Study**: This study is an experiment because of the manipulation of the independent variables.
- **Number of Factors**: This study follows a Multi-Factor Design because of the presence of multiple independent variables.

3. Apparatus and Materials

The experiment was conducted on a MacBook Air with an M1 chip, 8GB of RAM, running macOS Sequoia 15.1. The programming language used was OpenJDK 21.0.4, with VSCode version 1.92.1 as the integrated development environment (IDE). To ensure consistency and minimize interference, no other user processes were running in the background during the experiment.

4. Procedure

1. **Initialize Sorting Algorithms**:
   - Define an array of sorting algorithms to test, each implementing a `sort` method (e.g., `BubbleSortUntilNoChange`, `BubbleSortWhileNeeded`, `QuickSortGPT`, `SelectionSortGPT`).
2. **Define Datasets**:
   - Create datasets of varying sizes (100, 1,000, and 10,000) and data types (Integer, Long, Float, and Double).
   - For each data type, initialize arrays for the specified sizes.
3. **Generate Dataset Configurations**:
   - For each dataset, generate four initial configurations of data:
     - **Random**: Populate the array with randomly generated values.
     - **Reversed**: Populate the array with values in descending order.

    - **First-half-sorted**: Sort the first half of the array, with the remaining elements randomized.
    - **Last-half-sorted**: Sort the last half of the array, with the initial elements randomized.

4. **Warm-Up Phase**:
    - For each sorting algorithm, each dataset, and each sortedness level, perform an initial set of 25 sorting operations. These warm-up runs are discarded from the final results to allow the system and algorithm to stabilize.

5. **Measure Execution Time**:
    - For each sorting algorithm, dataset size, data type, and sortedness level, perform 100 timed sorting operations:
        - Use `System.nanoTime()` to measure the execution time for each sort.
        - Record the time taken in nanoseconds for each sort in a CSV file.

6. **Store Results**:
    - Record the algorithm name, data type, data size, sortedness level, and time taken for each run in the CSV file to allow for subsequent analysis.

7. **Analyze Data**:
    - Process the CSV file using python3.12.4 in a Jupyter Notebook to create graphs and tables, analyzing the relationship between independent variables (sorting algorithm, data size, data type, and sortedness level) and the dependent variable (execution time).

4. Results

1. Visual Overview

Provide an insightful overview of the data you collected. This requires some engineering from your part, to find a good degree of summarization: On one end of the spectrum, you don't summarize, and report hundreds of raw measurement values in a block of text. On the other end of the spectrum, you report a single number (like a mean value). Both approaches are bad.

Instead, use appropriate visual summaries (such as scatter plots, histograms, box plots, or empirical cumulative distribution functions) to show the distribution of your data. If you have a very small number of measurement values, then report all of them in a well organized table (where rows and/or columns correspond to different levels of different factors)

2. Descriptive Statistics

For each group or condition, summarize the set of measured values with a ”five-number summary”: minimum, first quartile, median, third quartile, and maximum.

Make sure you explain in your words what these statistics mean “in plain English”, but don't yet interpret them (this is for the Discussion section).

5. Discussion

1. Compare Hypotheses with Results

Provide a brief restatement of the main results from the previous section, and if (or if not) these support your research hypothesis.

If there is a discrepancy between your hypothesis and the results of your experiment, speculate about why you were unable to find evidence to support your hypothesis.

2. Limitations and Threats to Validity

Acknowledge any faults or limitations your study has, and how seriously these affect your results. How could these be remedied in future work?

3. Conclusions

End with the main conclusions that can be drawn from your study.

6. Appendix

1. Materials

Any documents you used for your informed consent (information sheets, consent) or as part of your apparatus (e.g., manual, hand-out), please include them here.

2. Reproduction Package

All of the code used to conduct the experiment, as well as the Jupyter Notebook used for data analysis and the Latex files for the report, can be found at the following GitHub repository: `https://github.com/costanza1234/USI-Exp-Eval-24`.