WAB

Provadis School of International Management and Technology

Comparing Suffix Automata Against Suffix Arrays For Longest Common Substring Queries

Rubin Chempananickal James
rubin.chempananickal-james@stud-provadis-hochschule.de
Matriculation Number: D876

Department: Information Technology
Module: Algorithmen und Datenstrukturen
Reviewer: Prof. Dr. Volker Scheidemann

February 21, 2026

Abstract

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

1.2 Research Questions

2 Methods

2.1 Design

2.2 Data Collection

AI Declaration

The usage of AI tools within this project is documented here. I solemnly declare that I have documented all interactions with AI tools, including the prompts used and the outputs received.

System	Prompt	Usage
GitHub Copilot 1	Asked to delete the existing project and provide a simple LaTeX template with a shared preamble, a Chapters subdirectory, TOC, glossary, abbreviations, and Roman numerals for non-content pages.	Template structure and LaTeX setup provided
GitHub Copilot 2	Requested additional title page lines (WAB header, reviewer, module) for the main document.	Title page metadata and layout updated
GitHub Copilot 3	Requested uppercase Roman numerals for front matter and lowercase Roman numerals for back matter page numbering.	Page numbering adjusted in main and exposee
GitHub Copilot 4	Requested exposee title page to include shared info fields (WAB header, department, module, reviewer).	Exposee title page updated to include shared metadata
GitHub Copilot 5	Requested adding the provadis-hochschule.pdf logo to the top right corner of both main and exposee title pages.	Logo added to top right of both title pages
GitHub Copilot 6	Reported that Glossary and Abbreviations sections were missing from compiled output.	Added example `\newacronym` entries to abbreviations.tex
GitHub Copilot 7	Reported that glossary section still not appearing in final PDF.	Updated settings.tex to include `automake` option in glossaries package to enable automatic glossary generation
GitHub Copilot 8	Requested Python benchmark code for LCS comparison with multiple synthetic scenarios (random strings, implanted mutated substrings, etc.)	Created benchmark script and documentation in Code folder

System	Prompt	Usage
GitHub Copilot 9	Asked to add a progress-bar-like output.	Added optional single-line in-place progress output with percentage, completed/total steps, elapsed time, ETA, and `-progress`/`-no-progress` flags
GitHub Copilot 10	Reported that ETA was inaccurate due to varying string lengths	Reworked ETA estimation to be algorithm- and length-aware using observed per-bucket runtimes
GitHub Copilot 11	Requested separate build and query timing (and plots).	Refactored benchmark to record build/query time separately and added total time summaries
GitHub Copilot 12	Requested plots using statistics beyond the mean (median, standard deviation, etc.).	Expanded summary stats and generated mean+std and median+IQR plots alongside memory plots
GitHub Copilot 13	Requested splitting benchmarking and plotting so plot generation can run separately, with a shared CLI entry point and default run-then-plot behavior.	Added CLI mode switch (run/plot/both) to reuse saved CSV results for plotting without rerunning benchmarks
GitHub Copilot 14	Requested an error if the two algorithms return different LCS result strings for the same input.	Added strict substring equality check (in addition to length) and raise an error on mismatch
GitHub Copilot 15	Requested that, in case of multiple valid LCS results, both algorithms return all results and compare the sets.	Updated both algorithms to return all max-length substrings and compare result sets for correctness
GitHub Copilot 16	Requested refactoring the benchmark script into helper modules under a helpers subdirectory, with separate algorithm imports and plotting helper, and clean top-level imports in the main file.	Split benchmark code into dedicated helper modules (SAM, ESA, plotting, generation, statistics, benchmarking) and simplified main script orchestration
GitHub Copilot 17	Requested mean, median, standard deviation, and IQR for memory usage plots as well.	Added memory quartile aggregation and generated memory mean+SD plus median+IQR plots
GitHub Copilot 18	Asked for a practical memory/space-complexity metric and whether build-time and query-time memory can be measured separately.	Added separate build/query phase memory metrics and a persistent index-size metric, with aggregation and plots for comparison
GitHub Copilot 19	Asked whether adding total time (build + query) columns and graphs with full statistics would make sense.	Added full total-time statistics (mean, median, std, IQR, Q1, Q3) to summaries and created dedicated total-time plots

System	Prompt	Usage
GitHub Copilot 20	Requested moving plot legends away from the top center because they obscured titles.	Repositioned all figure legends to the right side outside the plotting area to keep titles unobstructed
GitHub Copilot 21	Requested an additional output file containing both generated strings and their LCS value(s).	Added export of one CSV per generated case with ‘s‘, ‘t‘, LCS length, and all LCS substrings

Declaration of Authorship

I hereby confirm that I have personally and independently prepared the present work and have not used any sources or aids other than those specified. All passages taken verbatim or in substance from other sources are identified as such. The drawings, illustrations and tables in this work are created by me or have been provided with an appropriate source reference. This work has not been submitted by me to any other university in the same or similar form for the acquisition of an academic degree.

Frankfurt, February 21, 2026 . . . .
Rubin Chempananickal James