Assignment 6: Suffix Index

Helen Garabedian

March 14, 2025

1 Introduction

This assignment focuses on exploring and comparing three suffix index data structures: the suffix trie, suffix tree, and suffix array. I built each data structure for DNA sequences of varying lengths to evaluate their efficiency in terms of construction time, search speed, and memory usage. In genomics research, efficient string searching is crucial for processing large sequences and identifying important genetic markers. The results, presented in Figures 1 and 2, provide insight into how each data structure performs under different conditions. Figure 1 shows the time required to build each structure, while Figure 2 details the search times using a fixed query pattern. These findings help determine which structure is most suitable based on the performance trade-offs.

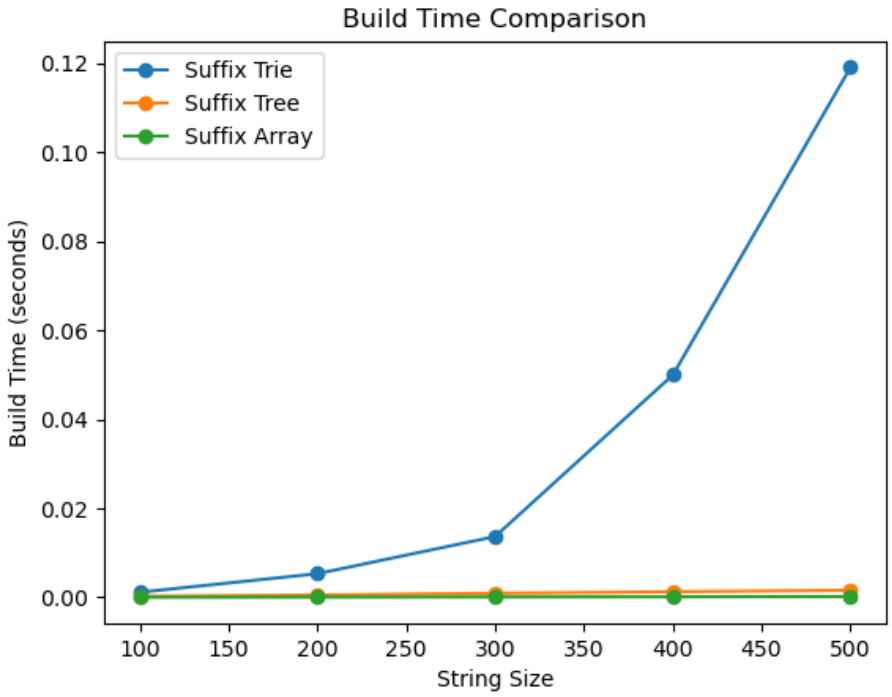

Figure 1: Build time comparison for the three suffix index data structures.

2 Results

The experimental results indicate that the suffix array generally offers a faster build process and lower memory usage compared to the suffix trie and suffix tree. Although the trie and tree structures provide direct search capabilities, they tend to incur higher memory overhead and longer build times, especially as the input string size increases. These quantitative measurements are clearly illustrated in the figures above, highlighting the trade-offs between build time, search efficiency, and memory consumption.

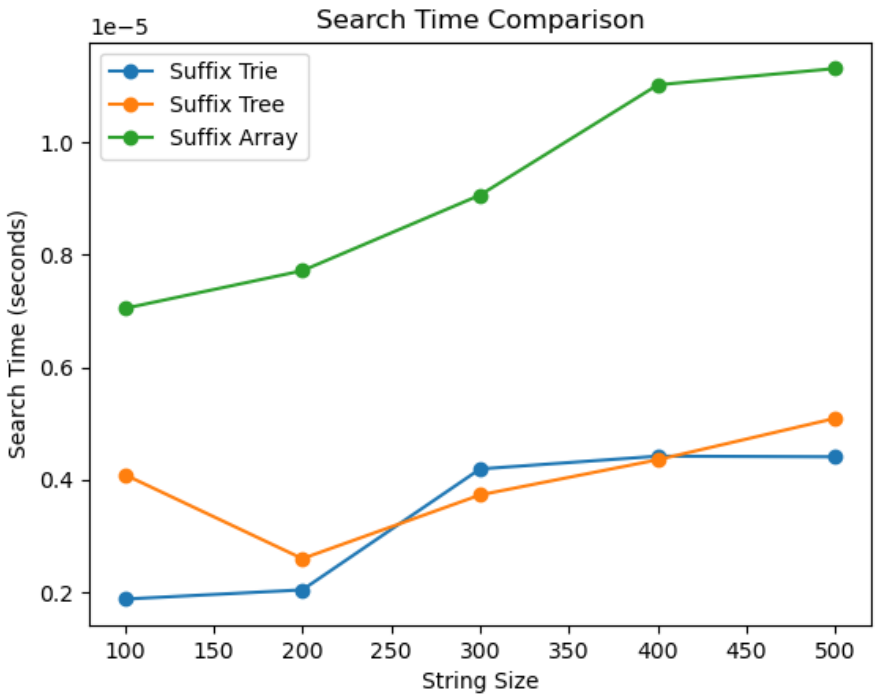

Figure 2: Search time comparison for the three suffix index data structures.

3 Methods

I generated random DNA sequences from the alphabet {A, C, G, T} over a range of string sizes to simulate realistic genomic data. For each sequence, I constructed the corresponding suffix trie, suffix tree, and suffix array, then performed search operations using a fixed query pattern. The elapsed time for both building and searching each data structure was recorded using Python's `time.perf_counter()`. Data visualization was accomplished with `matplotlib`, ensuring that the performance metrics are clearly presented and comparable across different structures. This methodical approach allowed me to capture a detailed performance profile for each data structure under controlled experimental conditions.

3.1 Reproducibility

```
$ git clone https://github.com/cu-compg-spring-2025/assignment-6-suffix-index-helengarabedian.git
$ cd assignment-6-suffix-index-helengarabedian
$ python experiment.py
```