Comparison of the Knuth–Morris–Pratt (KMP) and Rabin-Karp Algorithms for Pattern Matching

Course Code:	BLG 374E
CRN:	21352
Group Number:	23
Submission Date:	March 24nd, 2024

Member Name	**Member Student ID**
Yusuf Şahin	150200016
Mustafa Can Çalışkan	150200097
Mehmet Ali Balıkçı	150200059
Metin Ertekin Küçük	150210061
Yusuf Emir Sezgin	150200066

Table of Contents:

1. Introduction

Efficient pattern matching is crucial in data processing tasks such as text search and genome sequencing, where identifying patterns accurately amidst large datasets is challenging. Traditional methods often struggle with swift pattern discernment and detecting subtle variations, necessitating sophisticated algorithms like Knuth-Morris-Pratt, Boyer-Moore, Rabin-Karp, and Aho-Corasick. These algorithms, employing techniques like string matching, hashing, and finite automata, optimize search processes. By leveraging these diverse algorithms, practitioners can navigate through complex patterns, ensuring smoother data processing and accurate analysis across various domains.

2. Background

In the domain of information retrieval and computational analysis, the significance of robust pattern matching algorithms cannot be overstated, reminiscent of challenges encountered in diverse sectors. Similar to the identification of anomalies in financial transactions or the detection of irregularities in network traffic, pattern matching tasks demand meticulous attention to detail. Companies operating in data-intensive fields, such as cybersecurity firms or financial institutions, grapple with the complexities inherent in pattern recognition. Despite manual efforts to discern patterns, the sheer volume and intricacies of data often impede efficient processing. Thus, there arises an imperative to deploy advanced algorithms capable of swiftly and accurately identifying patterns, thereby enhancing operational efficiency and mitigating risks.

2.1. Literature Review

The studies of pattern matching find applications in numerous fields, spanning a wide array of domains. Some examples about research fields:

1. Bioinformatics: String matching algorithms play a vital role in bioinformatics, particularly in DNA, RNA, and protein sequence analysis[1].
2. Compressed String Matching: With the growth of data, compressed string matching, which involves searching for a pattern in compressed text, has become an important research area[2].
3. Text Mining and Natural Language Processing (NLP): String matching is crucial in text mining and NLP for tasks such as information retrieval, sentiment analysis, and machine translation[2].
4. String Matching in Databases: String matching is used in databases for query optimization and data retrieval[2].
5. Data Security: In the field of data security, string matching algorithms are used for intrusion detection within a network, identifying plagiarism, and digital forensics[3].
6. Pattern Recognition: String matching is essential in pattern recognition, which is used in image processing, speech recognition, and artificial intelligence[4].

3. Proposal Summary

In our term project, we will conduct an analysis of the Knuth-Morris-Pratt (KMP) and Rabin-Karp algorithms used in addressing the mentioned problems, and compare these algorithms based on their performance and suitability for usage.

4. Outcome

Tables and graphics for visualizing performance and efficiency analyses, along with code snippets implementing the algorithms and visualizations specifically prepared for illustrating the algorithms, will be utilized in both the project report and presentation.

5. Value

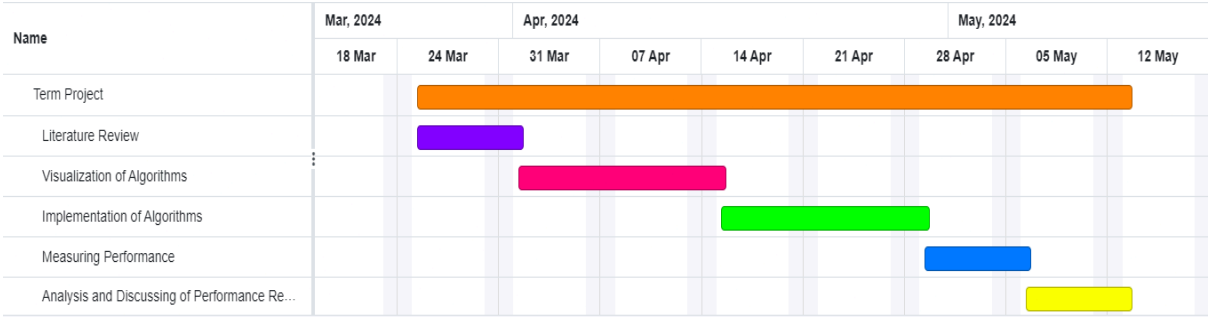
Comparing KMP and Rabin-Karp algorithms provides users with a better understanding of finding solutions to specific problems. This analysis gives a significant idea about which algorithm is more effective in which situations, providing insights into their application areas. Particularly used in various fields such as text processing, DNA sequence analysis, and database querying, understanding and using these algorithms correctly allows us to develop more effective solutions. Additionally, comparative analysis helps users choose the most suitable algorithm for a specific problem, preventing unnecessary time and resource consumption and providing more efficient solutions. In conclusion, comparing KMP and Rabin-Karp algorithms enables users to use these algorithms more effectively in their applications and projects, thus helping them develop more successful and innovative projects.

6. Methods

In this project, we aim to compare the KMP and Rabin-Karp algorithms, initially conducting a literature review to grasp their fundamental principles. We will then establish a test environment to implement both algorithms, evaluating their performance across texts of varying sizes. Our assessment will encompass performance metrics such as time complexity, memory usage, and operational counts, enabling a comprehensive comparison. Subsequently, we will analyze the findings to delineate the strengths and weaknesses of each algorithm. Finally, our results will be visually presented through tables, graphs, and code snippets in the report and presentation, facilitating a clear understanding and comparison of algorithmic performance. In this project, we will collaborate together and contribute jointly to all stages of the project.

7. Schedule

Presented below is the anticipated timeline delineating the phases of project tasks.



8. References

[1] : Pandiyarajan, Pandiselvam & T, Marimuthu & Raj, Lawrance. (2014). A Comparative Study on String Matching Algorithm of Biological Sequences.

[2] : Kari, L.; Konstantinidis, S.; Kopecki, S.; Yang, M. Efficient Algorithms for Computing the Inner Edit Distance of a Regular Language via Transducers. Algorithms 2018, 11(11), 165;

Hendrian, D.; Ueki, Y.; Narisawa, K.; Yoshinaka, R.; Shinohara, A. Permuted Pattern Matching Algorithms on Multi-Track Strings. Algorithms 2019, 12(4), 73;

Külekci, M.; Öztürk, Y. Applications of Non-Uniquely Decodable Codes to Privacy-Preserving High-Entropy Data Representation. Algorithms 2019, 12(4), 78;

Ghuman, S.; Giaquinta, E.; Tarhio, J. Lyndon Factorization Algorithms for Small Alphabets and Run-Length Encoded Strings. Algorithms 2019, 12(6), 124;

[3] : Mahmud, P., Rahman, A., Talukder, K.H. (2022). An Efficient Hashing Method for Exact String Matching Problems. In: Jacob, I.J., Kolandapalayam Shanmugam, S., Bestak, R. (eds) Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems. Springer, Singapore.

[4] : Understanding Pattern Matching. (2019, May 17). DeepAI.