The Boyer Moore Algorithm

Sachin Prasanna

Department of Information Technology
National Institute of Technology Karnataka, Surathkal

***Abstract* – This paper gives an overview on the Boyer Moore Algorithm, one of the powerful algorithms to perform string searching**

Key Terms – Boyer Moore Algorithm, Heuristics, Bad Character Rule, Good Suffix Rule

I. INTRODUCTION

The Boyer-Moore algorithm, developed by Robert S. Boyer and J Strother Moore, is a highly efficient string-searching algorithm widely used in computer science. It employs two key heuristics, namely the bad character rule and the good suffix rule, to optimize the search process and reduce unnecessary comparisons.

- The bad character rule allows the algorithm to shift the pattern based on the rightmost occurrence of a mismatched character, avoiding redundant comparisons.
- The good suffix rule leverages matching suffixes within the pattern itself to skip larger portions of the pattern during comparisons.

By combining these heuristics, the Boyer-Moore algorithm can quickly narrow down the search space and significantly improve the overall efficiency of the pattern matching process. These heuristics are complemented by the **precomputed shift tables**, including the bad character shift table and the good suffix shift table, which store the necessary shift amounts for each character and suffix, respectively.

II. PERFORMANCE ADVANTAGES

The Boyer-Moore algorithm offers excellent performance characteristics that make it highly efficient for pattern matching tasks. Its average-case time complexity is O(n/m), where n is the length of the text and m is the length of the pattern, making it particularly efficient when the pattern size is significant compared to the text.

The algorithm's efficiency is especially noticeable in scenarios with frequent mismatches or a high number of repetitive patterns, where it outperforms other algorithms such as the naive algorithm or the Knuth-Morris-Pratt algorithm. It is particularly well-suited for searching patterns in large texts or files, making it a popular choice in applications like text editors, compilers, etc. While the algorithm excels in many cases, patterns with many repeated characters may pose challenges where alternative algorithms like the Rabin-Karp algorithm may be more suitable.

III. SIMPLICITY AND IMPLEMENTATION

The Boyer-Moore algorithm's simplicity and ease of implementation contribute to its widespread use and adoption. It has a straightforward approach, utilizing shift tables and heuristics, simplifies the implementation process and allows for efficient handling of pattern matching tasks. The precomputed shift tables are constructed based on the pattern, enabling multiple searches with the same pattern to be performed without the need for recomputation.

The algorithm's simplicity also facilitates optimization and customization, leading to various enhancements and modifications over the years to further improve its efficiency and adapt it to specific use cases. Despite its simplicity, the Boyer-Moore algorithm has proven to be a powerful and effective tool for string searching, with applications in various fields.

IV. APPLICATIONS AND FUTURE SCOPE

Its development marked a breakthrough in efficient pattern matching, introducing novel heuristics and techniques that significantly improve performance and speed. The algorithm has inspired further research and advancements, leading to variations such as the Turbo Boyer-Moore algorithm and the Boyer-Moore-Horspool algorithm, which further optimize its efficiency in different scenarios. Beyond traditional computer science applications, the algorithm has found utility in bioinformatics for DNA sequence alignment. This hence aids in genetic research and medical advancements a lot.

Ongoing research aims to further optimize the algorithm's performance, explore its applications in new domains, and develop variations tailored to specific use cases, ensuring its continued relevance and growth.