

HOMEWORK REPORT

23050111029 Medine Merve MARAL

Sources

https://en.wikipedia.org/wiki/Boyer%E2%80%93Moore_string-search_algorithm
<https://stackoverflow.com/questions/27428605/constructing-a-good-suffix-table-understanding-an-example>
<https://www.geeksforgeeks.org/dsa/pattern-searching/>
<https://www.geeksforgeeks.org/dsa/boyer-moore-algorithm-for-pattern-searching/>
<https://www.geeksforgeeks.org/dsa/naive-algorithm-for-pattern-searching/>
<https://www.geeksforgeeks.org/dsa/kmp-algorithm-for-pattern-searching/>

I use all the sources below to refresh my knowledge and understand the Boyer-Moore algorithm. I was a little bit confused about how the bad character and the good suffix tables are created and work. Additionally, I searched for the algorithm's best cases for the pre-analysis method.

AI Chats

<https://gemini.google.com/share/fda580dda1d9> -> Boyer-Moore, the bad character and the good suffix rules

<https://chatgpt.com/share/693c60f8-15bc-8003-90f8-4297d2aee3cf>

<https://gemini.google.com/share/6a7020bd5d5b> -> Pre-Analysis idea

<https://gemini.google.com/share/a702dd596013> -> Using map for the bad character heuristic table

Boyer-Moore Implementation Approach

I implemented a classical bad character rule for Boyer-Moore string matching. Hash Map is used for the table so it can be used for any character set and special characters.

Pre-Analysis Strategy

For short texts and patterns, the algorithm chooses Naive because it's clearly the most efficient.

When the text has repetitive characters, the method chooses Rabin Karp if the alphabet is probably big or KMP, if the alphabet is probably smaller. For other situations, Boyer Moore is chosen.

isRepetitive method checks if a random character is repeating a lot in the text.

isAlphabetBig method takes the first `text.length/5` characters from the text and if these characters are repeating a lot, it means the alphabet is small.