Cüneyt Şahin - 22050111045
Berat Mut - 22050111069

CENG303 - DESIGN AND ANALYSIS OF ALGORITHMS
Bonus Homework-Brief Report

1 Boyer-Moore Implementation Approach:

We initially added Boyer-Moore with the badchars heuristic. However, one of the test cases wasn't working. Upon examining the test cases, we found that the problem occurred in Unicode tests. We suspected this might be due to limiting the alphabet size. When we asked the AI for the maximum alphabet size and added it, it passed all the cases, but the system slowed down considerably.Then we developed a system like this: If there was an ASCII value higher than 256 inside, we would write 65.536 as the alphabetical size, and if there wasn't one higher than 256, we would write 256. And only that test case was fast abroad, and that test case passed successfully.

2 GoCrazy Algorithm Design And Rationale:

For goCrazy, we came up with this idea: Boyer-Moore didn't seem fast enough for us with long patterns. While researching whether a faster version could be made with LLM, we found an algorithm called the Sunday algorithm. It seemed logical; normally, Boyer-Moore jumps by the length of the pattern, but this algorithm jumps by the pattern length plus 1. AI wrote this only for Unicode, and we sped it up a bit.For speeding up the process, we used arrays for ASCII characters and hashmaps for Unicode characters.

3 pre-analysis strategy :

In the pre-analysis phase, we tried many options with if statements, but mostly encountered poor results. Then we realized that the more if-else statements or other algorithms we added, the slower the system became. So we decided to check 2-3 basic things. We used naive, kmp, and our own goCrazy with 3 if loops. We didn't use Boyer Moore because goCrazy is already faster.

4 Analysis of your results :

We think the algorithm we wrote is quite fast. The test results show that. We saw that Naive was successful in so many test cases. In the analysis section we wrote, we observed that the red color appeared with very small speed differences.I think the analysis part we wrote is efficient. Actually, every algorithm is more successful in specific situations. This assignment helped us see this.

5 My Journey :

We found that the standard Boyer-Moore algorithm, which we usually write with 256 characters, doesn't work with Unicode characters. We spent many hours in the pre-analysis phase, but couldn't get all the values to turn green. It was difficult to see the increasing red color as the number of if/else statements or methods increased. Furthermore, we realized that writing our own algorithm without using an LLM was impossible. We couldn't find enough websites or information on the internet regarding this.Other than that, we liked the assignment. We thank our teacher Harun.

6 LLM link

`https://gemini.google.com/share/090ad9ceeebf`