NAME: Semon Yousri Samir Saleeb .

YEAR: third.

Subject: HUMAN GENOMICS.

The name of the paper: EMPIRICAL PERFORMANCE EVALUATION OF KNUTH MORRIS PRATT AND BOYER MOORE STRING MATCHING ALGORITHMS.

ABSTRACT:

1. Boyer Moore (BM) and Knuth Morris Pratt (KMP) algorithms have been proposed for string matching in order to find a specific pattern in a given text.
2. The aim of this paper is to evaluate the performance of two popular algorithms: Boyer Moore (BM) and Knuth Morris Pratt (KMP) in terms of execution time.
3. The algorithms have been programmed using Java and Java Microbenchmark Harness to evaluate their execution time using a number of experimental test scenarios
4. Results show that the BM algorithm outperformed the KMP algorithm in all test scenarios.

Introduction:

1. Nowadays, string matching algorithms which are widely used in most text editors, genetics, internet search engines, natural language processing, etc.
2. The main objective is to find all occurrences of a pattern (P) with size (M) in a text (T) of size (N), where (M<N) and P, T ∈ Σ, where Σ is a set of finite elements (symbols) taken from alphabet of a give language (called alphabet).

related work:

The new feature of this paper is that it compares the execution time­ of BM and KMP algorithms empirically

. Shows a substring search algorithm that outperforms the BM and Six separate matching algorithms were used to evaluate eight different texts of the same size from eight different natural languages

. Evaluates five string searching algorithms; Brute Force, BM, KMP, Karp-Rabin and the Horspool algorithm, through presenting how they work and decide which one of them is the best choice.

. presents a mixed string matching algorithm built on the KMP and BM string matching algorithms and that merging earning larger shift in case of mismatch.

. this paper differs from this works in using different text and using different pattern sizes or text language.

Boyer moore algorithm(bm):

The Boyer Moore one of the most effective algorithms for string search applications.

1. The algorithm start matching the character from right to left.
2. If it found a mismatch character it used one of the two method to shift the pattern depend on the method which gives the maximum number of shifts.
3. This two methods are [“bad character shift rule” and “good suffix shift rule”].
4. . The time complexity for pre-processing of a text with length n and pattern with length m is O(m + size of alphabet). The best case performance time complexity is O(n/m), and worst case time complexity is O(nm).

Bad character shift rule:

We start matching the pattern with the text from the write if all the pattern match then we found it but if their is a mismatch in a character then we search in the pattern about the text character which cause the mismatch .

Good suffix shift rule:

In this method we found a matched suffix in the right of the pattern and after it a mismatch character and instead of searching for one character we search in the pattern for another appearance to the matching suffix.

KNUTH MORRIS PRATT ALGORITHM (KMP):

The Knuth Morris Pratt algorithm was the first linear-time algorithm developed for the exact pattern matching problem.

1. The algorithm start matching the character from left to right.
2. If it found a mismatch character it used the longest proper prefix that is also suffix (LPS) table which is prepared before the comparison process started.
3. The time complexity for preprocessing of the pattern with length m can be done in O(m), and the worst case time complexity for searching phase is O(n + m).

the longest proper prefix that is also suffix (LPS):

It is one-dimensional array, with a number of elements equal to the number of characters in the pattern, its describe “How many positions the pattern has to shift” if there is a mismatch

METHODOLOGY:

It contain the test scenarios , conditions , metrics , setup.

Using this approach, the efficiency of the BM and KMP algorithms in terms of execution time has been empirically contrasted.

1. Test Scenarios: This test uses six text sizes and thee pattern size. And this selected to cover all aspects of each algorithms performance.
2. Test Conditions: first each algorithm tested with the same test scenario mentioned , the test program run in the same computer with closing all the program except it and without internet ,then we consider the time and Every test scenario has been recorded using the following JMH properties(Forks, Warm-up , Measurement , Threads , Benchmark mode)
3. Test Metrics: The time it takes to run the algorithms has been stated as a metric for evaluating and comparing the efficiency of both algorithms empirically. As a result, the algorithm with the shortest execution time is considered the highest in terms of efficiency.
4. Test Setup: it describe the software and the hardware specifications.

DISCUSSION:

All scenario tests show that The Boyer Moore much better than The Knuth Morris Pratt.

for text sizes 1 KB, 10 KB, 100 KB ,1 MB , over text size (100 MB) and over text size (200 MB) 🡪BM algorithm three time faster than KMP .

when the pattern size was (30) characters 🡪 show that the execution time of both algorithms have been decreased when the pattern size increased.🡪 the BM algorithm executed around (5) times faster than KMP algorithm when the pattern size was (30) characters

CONCLUSION:

This research Illustrates the performance of BM and KMP and this showed that BM algorithm is around (3) times faster than KMP algorithm and the pattern size did not effect on the performance very much.

In the future more research can be done in different languages or using different text and pattern sizes.