Comparison of various Substring Search Algorithms for a given input

Problem Statement:
Write a program that times the four methods for the task of searching for the substring "it is far far better thing that i do than i have ever done" in the text of Two cities (tale.txt). Discuss the extent to which your results validate the hypotheses about performance that are stated in the text.

Introduction:

The four algorithms that are mentioned in the problem statement include:
1. Brute Force Search
2. BoyerMoore Algorithm
3. Rabin Karp Algorithm
4. KMP Algorithm

Intuitively I would think that brute force would be the least efficient algorithm since it searches the given text exhaustively(O(N^2)) in order to find the pattern. Rabin Karp would take constant time as it uses a hash table to find the substring. Coming to BoyerMorre, it would hypothesize that its performance would be average amongst the four. But KMP, on the other hand, would be fastest in my opinion as it uses DFA simulation. A DFA is a smallest and fastest computational model one can build for a given problem. So the order of how fast the algorithms would be
KMP > BoyerMoore > Rabin Karp > Brute Force, KMP being the fastest and Brute Force being the slowest.

Procedure:

The Procedure I followed to conduct this experiment is as follows:
1. Write the code for each of the above algorithms in the context of the given tale.txt file and the pattern string.
2. I made use of the Files.readAllBytes() method from the java.nio package to do read the data in the text file and parse the data into a string.
3. I ran each and every algorithm once for the same output and measured how much time is it taking to run each of the algorithms

4. Since running the program only isn't sufficient to gather enough data for analysis, I ran each algorithm for 100 times and computed the average time taken for the execution and the standard deviation for the same data set.
5. I automated the above step using a simple python script and then plotted the values as a scatter plot to visualize the results.

Analysis:

Here is a table of the results that I have obtained after said experimentation.

Name of the Algorithm	Average Time of Execution (MilliSeconds)	Standard Deviation (MilliSeconds)
BoyerMoore	28	1.380
KMP	33	1.980
Brute Force	35	2.180
Rabin Karp	68	2.120

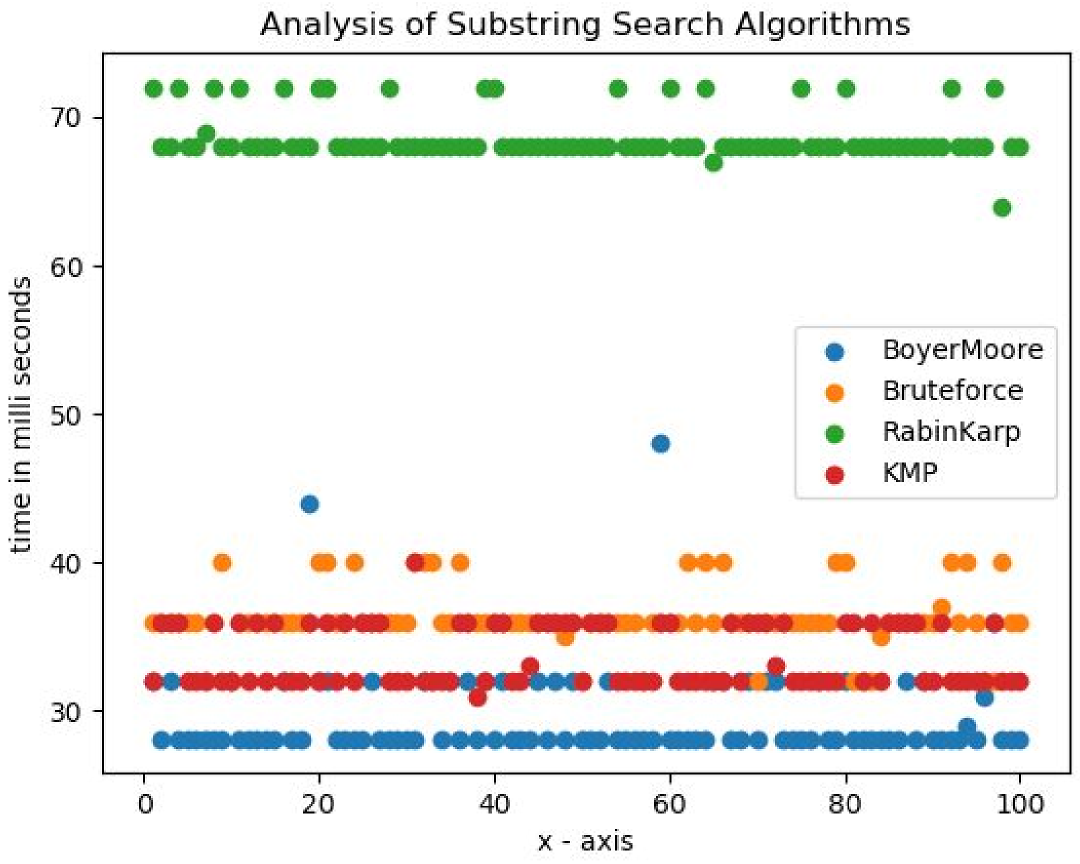
We can clearly observe that BoyerMoore is the fastest among the four, it goes on to say that even the DFA model approach to solving this problem isn't fast enough and one more observation that we can make is that brute force was supposed to be the worst performer, and we can clearly observe that it isn't.

The explanation of why Rabin Karp is the slowest can be attributed to the hash function that is written, in this case, the function that is written isn't a good thereby reducing the constant time required to the worst time of the four algorithms.

KMP is a linear time algorithm and it is found to be linear during the experimentation.

In the end, BoyerMorre is the fastest thanks to its mismatch character skipping heuristic.

Here is the graph that I have generated to visualize this data



Conclusion:

Boyer-Moore Algorithm is the fastest substring search algorithm for the given data set and use case.