

### Related Work :

Brute Force is a primary method which preprocesses neither the text nor the pattern. From left to right, BF performs a character by character analysis. The sliding window is moved one place to the right after a match or mismatch, and the matching is restarted from the first character of the pattern. A major drawback of BF is its high time consumption. There are approaches that combine the dynamic programming approach and DFA that are based on Deterministic Finite Automata. These methods aren't always scalable for large sequences due to the use of a finite automaton. Furthermore, because of the use of dynamic programming, the memory requirements are much higher. The KMP algorithm was proposed by Knuth which performs the comparison from the left side. When a mismatch occurs, KMP pushes the sliding window to the right by keeping the longest overlap of the matched text's suffix and the pattern's prefix. The output of this algorithm is linear. The KMP algorithm works well when the alphabet size is high, but it takes a long time to run when the alphabet size is small or the pattern length is short. From right to left, the Boyer-Moore algorithm looks for a pattern in the text. This algorithm looks for the pattern's last character first. It computes the shift increment at the end of the matching phase. When a mismatch arises, two helpful rules (bad character and good suffix) are used to reduce the number of comparisons. The Boyer-Moore algorithm's drawback is that its preprocessing time is dependent on the pattern length and alphabet size. The Divide and Conquer Pattern Matching algorithm is based on comparisons. The text is checked for the pattern's rightmost character at the start of the DCPM's preprocessing phase. The rightmost character table contains the index of the findings. The text is scanned once more to detect pattern's leftmost character. The indexes are saved in the leftmost character table in the case of sameness. DCPM determines the window boundaries by using these two tables. In other words, the elements of the tables are investigated based on the length of the pattern. When the distance between the windows' leftmost and rightmost character equals the pattern length, a window is found. As a result, DCPM requires two passes through the text as well as some calculations to decide the windows. Total sameness is achieved when all of the pattern's characters and the text's windows are identical.