

In the pattern matching algorithm, Brute Force (BF) is a primary method which preprocesses neither the text nor the pattern. BF compares character by character from left to right. After either a match or mismatch, it is shifted one position to the right and the matching is restarted from the first character of the pattern. But the disadvantage of BF is that it consumes a lot of time. There are methods based on Deterministic Finite Automata (DFA) that combine the dynamic programming approach and DFA. Due to the use of a finite automaton, these methods are not often scalable for large sequences. In addition, the memory requirements are greater because of the usage of dynamic programming. The KMP algorithm which performs the comparison from the left side. If a mismatch happens, it moves to the right by holding the longest overlap of a suffix of the matched text and a prefix of the pattern. Although it performs well when the alphabet size is large, the KMP algorithm requires a long run time when either the alphabet size is small or the length of the pattern is short. The Boyer-Moore first matches the pattern's last character. At the end of the matching phase, it computes the shift increment. To decrease the number of comparisons when a mismatch occurs, two useful rules (bad character and good suffix) are utilized. The disadvantage of the Boyer-Moore algorithm is the dependency of its preprocessing time on the pattern length and alphabet size.