Abstract and Introduction summary :

- The pattern matching is a very practical operation ,its enables users to find the locations of particular DNA subsequences in a database or DNA sequence so , high-speedpattern matching algorithms are needed .
- The problem is, biological data are enormous in size and very complex, we need efficient ways to analyze the data such that Human genome contains approximately 3 billion base pairs (high-computational costs) .
- The experimental results demonstrate the superiority of the presented algorithms over the other simulated algorithms and the development of efficient algorithms is still required .
- Pattern matching algorithms play a vital role in computational biology , Its necessary to study ourselves and learn about variant characteristics .
- The present paper introduces three pattern matching algorithms that are specially formulated to speed up searches on large DNA sequences , focuses on the exact pattern matching problem which finds all the occurrences of a pattern in a text .
- In the pattern matching problem, a text, sequence or database is scanned to detect the locations of a pattern in the text .
- The pattern matching problem arises in the different scopes of computational bioinformatics, which include the basic local alignment search, biomarker discovery, sequence alignment, proteogenomic mapping, and homologous series detection. In these disciplines, there is a need to recognize the locations of multiple patterns, including those of amino acids and nucleotides in databases .
- The comparison of a particular gene with similar genes of the same or different organisms and the prediction of its function .
- In another application, the functionality of a recently discovered DNA sequence can be prespecified by investigating its similarity to known sequences of DNA. This approach has been used in various research studies and medical applications.
- The operation of the proposed algorithms is divided into a preprocessing and a matching phase. In the preprocessing phase, the potential intervals of the text to be matched with the pattern are recognized , These intervals are called windows , during the matching phase, the windows are carefully scanned in order to be matched with the pattern .
- The fewer windows found in the preprocessing phase, the less time taken for verifying the windows in the matching phase , the present work's primary aim is to decrease the number of recognized windows .
- The present study introduces a third algorithm that focuses on the word of the pattern having the fewest repetitions in the text. In other words, the algorithm searches the text for a low-frequency word of the pattern. This technique further advances the algorithm's efficiency by decreasing the number of discovered windows.
- The KMP algorithm which performs the comparison from the left side. In the event of a mismatch, KMP moves the sliding window to the right by holding the longest overlap of a suffix of the matched text and a prefix of the pattern , This algorithm has a linear performance.
- The Boyer-Moore algorithm and its variants search the pattern in the text from right to left. In other words, this algorithm 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 Divide and Conquer Pattern Matching (DCPM) is a comparison-based algorithm. At the beginning of the DCPM's preprocessing phase, the text is scanned for the rightmost character of the pattern. The index of the findings is stored in the rightmost character table. Then, to detect the leftmost character of the pattern, the text is scanned again.
  In the case of sameness, the indexes are saved in the leftmost character table.