String Matching Algorithms

String Matching Algorithms

1. **Definition**
    - Algorithms that find occurrences of a pattern string within a larger text string.
2. **Naive String Matching**
    - Checks every possible starting position in the text for a match, with worst-case time complexity O(nm) for text length n and pattern length m.
3. **Knuth-Morris-Pratt (KMP) Algorithm**
    - Uses a precomputed "failure function" to avoid re-examining characters already matched, achieving O(n+m) time complexity.
4. **Rabin-Karp Algorithm**
    - Uses hashing to quickly compare substrings, particularly efficient for searching multiple patterns simultaneously.
5. **Boyer-Moore Algorithm**
    - Scans the pattern from right to left and can skip large sections of text, often the fastest in practice for single-pattern searches.
6. **Applications**
    - Text editors (find/replace), plagiarism detection, DNA sequence analysis, and search engines all rely heavily on efficient string matching.