**String Matching and Palindromic Substring Algorithms in JavaScript:**

String matching and palindromic substring algorithms are essential techniques used to search for patterns or structures within strings. JavaScript provides various algorithms for efficiently performing these tasks.

**String Matching Algorithms:**

Naive String Matching:

Naive string matching is a simple algorithm that iterates through the text and pattern strings to find occurrences of the pattern within the text. It checks whether the characters in the pattern match the corresponding characters in the text at each position. Naive string matching has a time complexity of O(m \* n), where m is the length of the pattern and n is the length of the text.

Knuth-Morris-Pratt (KMP) Algorithm:

The Knuth-Morris-Pratt algorithm is an efficient string matching algorithm that avoids unnecessary character comparisons by utilizing information about the pattern itself. It preprocesses the pattern to construct a partial match table (also known as the failure function) and uses it to skip ahead in the text when a mismatch occurs. KMP algorithm has a time complexity of O(m + n), where m is the length of the pattern and n is the length of the text.

**Longest Palindromic Substring Algorithm:**

Manacher's Algorithm:

Manacher's algorithm is an efficient algorithm used to find the longest palindromic substring within a given string. It preprocesses the string to transform it into a new string with special characters inserted between adjacent characters, allowing for efficient detection of palindromes. Manacher's algorithm has a time complexity of O(n), where n is the length of the input string.