ANALYSIS OF TIME COMPLEXITY

Terminologies:

Palindrome:

A palindrome is a sequence of characters that reads the same forwards and backward. For example, "radar" and "level" are palindromes.

Longest Palindromic Substring:

This refers to the longest substring of a given string that is a palindrome. For instance, in the string "babad", the longest palindromic substring is "bab" or "aba".

Detailed Complexity Analysis:

Preprocessing:

In this step, the algorithm preprocesses the input string to handle both odd and even-length palindromes efficiently. It inserts special characters (such as '#') between each pair of characters and at the beginning and end of the string.

This preprocessing step takes linear time, O(n), where n is the length of the original input string. This is because each character of the input string is processed once.

Main Algorithm:

The core of Manacher's Algorithm involves iterating through the processed string, centering at each character, and expanding outwards to find the longest palindromic substring centered at that character.

While iterating, the algorithm efficiently utilizes previously computed information to avoid redundant calculations. It maintains information about the lengths of palindromes centered at each position.

This main algorithm step also takes linear time, O(n), where n is the length of the processed string. Again, each character of the processed string is visited once.

Total Complexity:

Combining the preprocessing step and the main algorithm, the total time complexity of Manacher's Algorithm is O(n), where n is the length of the original input string.

This linear time complexity makes Manacher's Algorithm highly efficient for finding the longest palindromic substring compared to naive approaches, which would typically have time complexity closer to O(n^2) due to exhaustive checks.

Conclusion:

Manacher's Algorithm is an efficient algorithm for finding the longest palindromic substring in a given string. Its linear time complexity, O(n), where n is the length of the input string, makes it particularly suitable for real-world applications where performance is crucial. By cleverly utilizing previously computed information and employing dynamic programming-like techniques, Manacher's Algorithm achieves this efficient time complexity.