Z ALGORITHM

The Z algorithm exhibits linear time complexity (O(n)) in all cases (best, worst, and average) for constructing the Z array. Here's why:

Linear Time Complexity Explained

The time complexity of an algorithm refers to the amount of time it takes to execute, typically measured in terms of the size of the input data (n for the string length in the Z algorithm). We analyze best, worst, and average cases to understand the algorithm's behavior under different input scenarios.

Why Z Algorithm is Always O(n):

Fixed-Cost Operations: The Z algorithm primarily involves operations with a constant time cost, regardless of the input string length. These include:

Comparing characters

Copying values in the Z array

Updating variables (like l and r pointers)

Bounded Loop Iterations: The core loop iterates from i = 1 to i = n-1, guaranteeing that each element in the Z array is visited exactly once.

Limited Substring Comparisons: Even the worst-case scenario within the loop involves comparing characters up to the remaining string length (n - i). However, this comparison happens at most once for each element i in the Z array. Since the loop iterates n-1 times, the total number of character comparisons remains bounded by O(n) in the worst case.

Breakdown of Best, Worst, and Average Cases

Best Case: There isn't a specific "best case" scenario for the Z algorithm in terms of time complexity. All input strings will lead to the same linear time behavior.

Worst Case: The worst case occurs when the string has very few repeated prefixes. In this case, the brute-force matching inside the loop might compare more characters for some i values. However, as explained above, the total comparisons are still bounded by O(n).

Average Case: The average case also reflects the linear time complexity. The Z algorithm performs a similar number of operations on average for any random string, resulting in an average time complexity of O(n).

In essence, the Z algorithm's reliance on fixed-cost operations, a bounded number of loop iterations, and limited character comparisons ensure that its time complexity remains linear (O(n)) for all input strings, regardless of the pattern of repetitions within them.