**Time Complexity Analysis**

The time complexity analysis of the Rabin-Karp algorithm involves understanding the operations performed at each step and how they contribute to the overall runtime. Here's a detailed breakdown of the time complexity of the Rabin-Karp algorithm:

**1.** **Calculating Hash Values:**

- Computing the hash value of the pattern: O(m), where m is the length of the pattern.

- Calculating the hash value of the first substring of text: O(m), where m is the length of the pattern.

**2.** **Sliding the Pattern:**

- Iterating through the text and sliding the pattern: O(n-m+1), where n is the length of the text and m is the length of the pattern.

**3. Comparing Hash Values:**

- If hash values match, performing a character-by-character comparison: O(m), where m is the length of the pattern.

**4. Updating Hash Values:**

- Updating the hash value for the next substring: O(1).

Considering the above steps, the total time complexity of the Rabin-Karp algorithm can be approximated as follows:

- **Worst-case scenario** : O((n - m + 1) \* m) for a complete pattern matching for all substrings in the text. This arises when all substrings have the same hash value as the pattern, necessitating character-by-character verification.

- **Average-case scenario**: O(n + m) when the rolling hash function effectively reduces the number of character comparisons required.