**Rabin Karp Algorithm**

The time complexity of the Rabin-Karp algorithm for string searching can be analyzed as follows:

1. Preprocessing (Hash Calculation):

- Calculating the hash value for both the pattern and the initial window of the text takes O(m) time, where m is the length of the pattern.

2. Main Loop (Sliding Window):

- The main loop iterates over each character of the text once. In each iteration:

- Checking the equality of hash values takes O(1) time.

- If the hash values are equal, a full string comparison of length m (the pattern length) is performed, which takes O(m) time in the worst case.

- Recalculating the hash for the next window takes O(1) time.

- Therefore, the total time complexity for the main loop is O(n-m+1) \* O(m), where n is the length of the text.

- In the worst case, when the pattern matches at every position in the text, this simplifies to O(nm).

Overall, the time complexity of the Rabin-Karp algorithm is O(nm) in the worst case and O(n) on average when considering the hash calculation and comparison operations. However, its average-case performance can degrade to O(nm) if hash collisions are frequent.