Time Complexities of Rabin-Karp Algorithm:

The provided code implements the Rabin-Karp algorithm for string searching. Let's break down the time complexities for different cases:

**Best Case:**

* The best case occurs when the pattern is found at the beginning of the text.
* In this case, the time complexity is O(N), where N is the length of the text.
* This happens because after the preprocessing step where the hash values for both the pattern and the first substring of the text are calculated, only a single comparison is needed to confirm a match.

**Worst Case:**

* The worst case occurs when all substrings of the text need to be checked against the pattern.
* In this case, the time complexity is O(N×M), where N is the length of the text and M is the length of the pattern.
* This worst-case scenario arises when the hash function generates hash collisions for many substrings, leading to the need for character-by-character comparisons.

**Exceptional Case:**

* An exceptional case can arise if there are many hash collisions even when the pattern is not present in the text.
* Although the Rabin-Karp algorithm typically has good average-case behavior, in the exceptional case where there are numerous hash collisions, the time complexity can degrade.
* In such cases, the algorithm may need to perform character comparisons for many substrings, leading to a time complexity similar to the worst case.
* However, the likelihood of this exceptional case is generally low, especially with a well-designed hash function and a sufficiently large prime number q chosen for hashing.

In summary, the Rabin-Karp algorithm has a best-case time complexity of O(N), a worst-case time complexity of O(N×M), and an exceptional case time complexity that can approach the worst case in scenarios with frequent hash collisions.