1. Binary Sunday vs. Gusfield Z

Results:

Sunday: 127.1367073059082 microseconds, Gusfield Z: 1285.2120399475098 microseconds

Explanation:

- Binary Sunday: This algorithm is designed to be efficient by making use of a shift table that allows it to skip sections of the text based on the character immediately following the current window. The time complexity is O(n\*m) for random texts, making it very efficient for searching short patterns in large texts.

- Gusfield Z: The Gusfield Z algorithm involves an initial preprocessing step to compute the Z-array, which takes O(n + m) time. This preprocessing can be quite costly, especially when the pattern is short and the text is large. The Z-array computation adds a significant overhead compared to the direct search performed by Binary Sunday.

Why Binary Sunday is Faster:

- Overhead: The Gusfield Z algorithm's preprocessing step creates an overhead that can be significant for short patterns, while Binary Sunday's approach directly processes the text with minimal setup.

- Skipping Efficiency: Binary Sunday effectively skips non-matching parts of the text based on the character after the current window, making it faster in practice for many cases.

2. KMP vs. Rabin-Karp

Results:

KMP: 91773.98681640625 microseconds

Rabin Karp: 183428.76434326172 microseconds

Explanation:

- KMP (Knuth-Morris-Pratt): KMP uses the LPS (Longest Prefix Suffix) array to preprocess the pattern. This array helps in skipping redundant comparisons by reusing previous match information. The preprocessing step takes O(m) time, and the search phase takes O(n) time, making it efficient for patterns with repetitive prefixes.

- Rabin-Karp: This algorithm uses hashing to find the pattern in the text. The preprocessing and searching both take O(n \* m) time in the worst case due to hash collisions, though it often performs better in practice.

Why KMP is Faster:

- Hash Collisions: Rabin-Karp can suffer from hash collisions, which necessitate additional comparisons, making it slower in practice when collisions are frequent.

- Efficient Skipping: KMP efficiently skips over non-matching sections by using the LPS array, reducing the number of comparisons needed.

3. Rabin-Karp vs. Binary Sunday

Results:

Rabin Karp: 73.28987121582031 microseconds, Sunday: 42672.11675643921 microseconds

Explanation:

- Rabin-Karp: As mentioned, Rabin-Karp uses hashing to speed up the matching process. When there are few hash collisions, the algorithm can skip large sections of the text efficiently. The average-case time complexity is O(n + m), but it can degrade to O(n \* m) in the worst case.

- Binary Sunday: This algorithm, as previously explained, uses a shift table to skip sections of the text based on the character following the current window, making it very efficient for most cases.

Why Rabin-Karp is Faster:

- Minimal Hash Collisions: In scenarios where hash collisions are minimal, Rabin-Karp can efficiently skip large portions of the text, leading to faster search times compared to Binary Sunday, which may have to process more characters.

- Efficient Hashing: The hashing mechanism allows for quick checks across the text, reducing the need for character-by-character comparisons.

Summary of the Comparisons:

1. Binary Sunday vs. Gusfield Z: Binary Sunday is faster due to the lack of preprocessing overhead and efficient skipping mechanisms.

2. KMP vs. Rabin-Karp: KMP is faster because it avoids redundant comparisons through the LPS array, while Rabin-Karp can be slowed down by hash collisions.

3. Rabin-Karp vs. Binary Sunday: Rabin-Karp can be faster when hash collisions are minimal, allowing it to skip large sections of the text efficiently.