### Algorithm: KMP (Knuth-Morris-Pratt) Pattern Matching

Sir Cedric and Ember needed to decode an ancient scroll that contained clues to Princess Elara's location. The scroll was written in a mysterious pattern that required the KMP algorithm to decipher.

#### Build LPS Array:

* Sir Cedric used his sword's runes (LPS array) to preprocess the scroll's pattern.

#### Decode the Scroll:

* He used the runes to match the pattern in the scroll efficiently.

#### Implementation:

| **def** kmp\_search(pattern: str, scroll: str) -> List[int]:  **def** build\_lps(pattern):  lps = [0] \* len(pattern)  length = 0  i = 1  **while** i < len(pattern):  **if** pattern[i] == pattern[length]:  length += 1  lps[i] = length  i += 1  **else**:  **if** length != 0:  length = lps[length - 1]  **else**:  lps[i] = 0  i += 1  **return** lps  lps = build\_lps(pattern)  results = []  i = j = 0  **while** i < len(scroll):  **if** pattern[j] == scroll[i]:  i += 1  j += 1  **if** j == len(pattern):  results.append(i - j)  j = lps[j - 1]  **elif** i < len(scroll) **and** pattern[j] != scroll[i]:  **if** j != 0:  j = lps[j - 1]  **else**:  i += 1  **return** results  *# Example usage:*  scroll = "ababcabcabababd"  pattern = "ababd"  print(kmp\_search(pattern, scroll)) *# Output: [10]* |
| --- |

#### Explanation:

Build LPS Array:

* lps: Sir Cedric used his sword's runes to preprocess the pattern.

Decode the Scroll:

### He used the runes to match the pattern in the scroll efficiently.