Given a **long text document (`text`)** and a **keyword/pattern (`pattern`)**, your task is to design an efficient algorithm to **find all occurrences of the pattern in the text**.

The algorithm must be highly optimized to handle cases where:

- The `text` is very large (up to `100,000` characters).
- The `pattern` can be moderately sized (up to `10,000` characters).
- The pattern may have repeating sub-patterns that could be exploited for optimization.

🚦 Input:

1. A string `text` representing the text document.
2. A string `pattern` representing the keyword/pattern to be searched.

🎯 Output:

- A **list of starting indices** (0-indexed) where the pattern is found in the text.
- If the pattern is not found, return an **empty list**.

Additional Requirement:

- **Your algorithm should run in O(n + m) time complexity**, where `n` is the length of the text and `m` is the length of the pattern.