Problem 3455: Shortest Matching Substring

Problem Information

Difficulty: Hard
Acceptance Rate: 22.89%
Paid Only: No
Tags: Two Pointers, String, Binary Search, String Matching

Problem Description

You are given a string `s` and a pattern string `p`, where `p` contains **exactly two** `'*'` characters.

The `'*'` in `p` matches any sequence of zero or more characters.

Return the length of the **shortest** substring in `s` that matches `p`. If there is no such substring, return -1.

Note: The empty substring is considered valid.

Example 1:

Input: s = "abaacbaecebce", p = "ba*c*ce"

Output: 8

Explanation:

The shortest matching substring of `p` in `s` is `"_**ba**_ e _**c**_ eb _**ce**_ "`.

Example 2:

Input: s = "baccbaadbc", p = "cc*baa*adb"

Output: -1

Explanation:

There is no matching substring in `s`.

Example 3:

Input: s = "a", p = "**"

Output: 0

Explanation:

The empty substring is the shortest matching substring.

Example 4:

Input: s = "madlogic", p = "*adlogi*"

Output: 6

Explanation:

The shortest matching substring of `p` in `s` is `"**_adlogi_** "`.

Constraints:

* `1 <= s.length <= 105` * `2 <= p.length <= 105` * `s` contains only lowercase English letters.
* `p` contains only lowercase English letters and exactly two `'*'`.

Code Snippets

C++:

```
class Solution {
public:
int shortestMatchingSubstring(string s, string p) {

}
```

```
};
```

Java:

```
class Solution {
public int shortestMatchingSubstring(String s, String p) {

}
}
```

Python3:

```
class Solution:
def shortestMatchingSubstring(self, s: str, p: str) -> int:
```