Problem 3135: Equalize Strings by Adding or Removing Characters at Ends

Problem Information

Difficulty: Medium
Acceptance Rate: 55.82%
Paid Only: Yes
Tags: String, Binary Search, Dynamic Programming, Sliding Window, Hash Function

Problem Description

Given two strings `initial` and `target`, your task is to modify `initial` by performing a series of operations to make it equal to `target`.

In one operation, you can add or remove **one character** only at the _beginning_ or the _end_ of the string `initial`.

Return the **minimum** number of operations required to _transform_ `initial` into `target`.

Example 1:

Input: initial = "abcde", target = "cdef"

Output: 3

Explanation:

Remove `'a'` and `'b'` from the beginning of `initial`, then add `'f'` to the end.

Example 2:

Input: initial = "axxy", target = "yabx"

Output: 6

Explanation:

Operation | Resulting String ---|--- Add `'y'` to the beginning | `"yaxxy"` Remove from end | `"yaxx"` Remove from end | `"yax"` Remove from end | `"ya"` Add `'b'` to the end | `"yab"` Add `'x'` to the end | `"yabx"` **Example 3:**

Input: initial = "xyz", target = "xyz"

Output: 0

Explanation:

No operations are needed as the strings are already equal.

Constraints:

* `1 <= initial.length, target.length <= 1000` * `initial` and `target` consist only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int minOperations(string initial, string target) {

}
};
```

Java:

```
class Solution {
public int minOperations(String initial, String target) {

}
}
```

Python3:

```
class Solution:
def minOperations(self, initial: str, target: str) -> int:
```