Problem 1208: Get Equal Substrings Within Budget

Problem Information

Difficulty: Medium
Acceptance Rate: 59.29%
Paid Only: No
Tags: String, Binary Search, Sliding Window, Prefix Sum

Problem Description

You are given two strings `s` and `t` of the same length and an integer `maxCost`.

You want to change `s` to `t`. Changing the `ith` character of `s` to `ith` character of `t` costs `|s[i] - t[i]|` (i.e., the absolute difference between the ASCII values of the characters).

Return _the maximum length of a substring of_`s` _that can be changed to be the same as the corresponding substring of_`t` _with a cost less than or equal to_`maxCost`. If there is no substring from `s` that can be changed to its corresponding substring from `t`, return `0`.

Example 1:

Input: s = "abcd", t = "bcdf", maxCost = 3 **Output:** 3 **Explanation:** "abc" of s can change to "bcd". That costs 3, so the maximum length is 3.

Example 2:

Input: s = "abcd", t = "cdef", maxCost = 3 **Output:** 1 **Explanation:** Each character in s costs 2 to change to character in t, so the maximum length is 1.

Example 3:

Input: s = "abcd", t = "acde", maxCost = 0 **Output:** 1 **Explanation:** You cannot make any change, so the maximum length is 1.

Constraints:

* `1 <= s.length <= 105` * `t.length == s.length` * `0 <= maxCost <= 106` * `s` and `t` consist of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int equalSubstring(string s, string t, int maxCost) {

}
};
```

Java:

```
class Solution {
public int equalSubstring(String s, String t, int maxCost) {

}
}
```

Python3:

```
class Solution:
def equalSubstring(self, s: str, t: str, maxCost: int) -> int:
```