Problem 3361: Shift Distance Between Two Strings

Problem Information

Difficulty: Medium
Acceptance Rate: 53.16%
Paid Only: No
Tags: Array, String, Prefix Sum

Problem Description

You are given two strings `s` and `t` of the same length, and two integer arrays `nextCost` and `previousCost`.

In one operation, you can pick any index `i` of `s`, and perform **either one** of the following actions:

* Shift `s[i]` to the next letter in the alphabet. If `s[i] == 'z'`, you should replace it with `'a'`. This operation costs `nextCost[j]` where `j` is the index of `s[i]` in the alphabet. * Shift `s[i]` to the previous letter in the alphabet. If `s[i] == 'a'`, you should replace it with `'z'`. This operation costs `previousCost[j]` where `j` is the index of `s[i]` in the alphabet.

The **shift distance** is the **minimum** total cost of operations required to transform `s` into `t`.

Return the **shift distance** from `s` to `t`.

Example 1:

Input: s = "abab", t = "baba", nextCost = [100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], previousCost = [1,100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

Output: 2

Explanation:

* We choose index `i = 0` and shift `s[0]` 25 times to the previous character for a total cost of 1. * We choose index `i = 1` and shift `s[1]` 25 times to the next character for a total cost of 0. * We choose index `i = 2` and shift `s[2]` 25 times to the previous character for a total cost of 1. * We choose index `i = 3` and shift `s[3]` 25 times to the next character for a total cost of 0.

Example 2:

Input: s = "leet", t = "code", nextCost = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1], previousCost = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]

Output: 31

Explanation:

* We choose index `i = 0` and shift `s[0]` 9 times to the previous character for a total cost of 9. * We choose index `i = 1` and shift `s[1]` 10 times to the next character for a total cost of 10. * We choose index `i = 2` and shift `s[2]` 1 time to the previous character for a total cost of 1. * We choose index `i = 3` and shift `s[3]` 11 times to the next character for a total cost of 11.

Constraints:

* `1 <= s.length == t.length <= 105` * `s` and `t` consist only of lowercase English letters. * `nextCost.length == previousCost.length == 26` * `0 <= nextCost[i], previousCost[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long shiftDistance(string s, string t, vector<int>& nextCost,
vector<int>& previousCost) {

}
};
```

Java:

```
class Solution {
public long shiftDistance(String s, String t, int[] nextCost, int[]
previousCost) {

}
}
```

Python3:

```
class Solution:
def shiftDistance(self, s: str, t: str, nextCost: List[int], previousCost:
List[int]) -> int:
```