Problem 3146: Permutation Difference between Two Strings

Problem Information

Difficulty: Easy
Acceptance Rate: 87.54%
Paid Only: No
Tags: Hash Table, String

Problem Description

You are given two strings `s` and `t` such that every character occurs at most once in `s` and `t` is a permutation of `s`.

The **permutation difference** between `s` and `t` is defined as the **sum** of the absolute difference between the index of the occurrence of each character in `s` and the index of the occurrence of the same character in `t`.

Return the **permutation difference** between `s` and `t`.

Example 1:

Input: s = "abc", t = "bac"

Output: 2

Explanation:

For `s = "abc"` and `t = "bac"`, the permutation difference of `s` and `t` is equal to the sum of:

* The absolute difference between the index of the occurrence of `"a"` in `s` and the index of the occurrence of `"a"` in `t`. * The absolute difference between the index of the occurrence of `"b"` in `s` and the index of the occurrence of `"b"` in `t`. * The absolute difference between the index of the occurrence of `"c"` in `s` and the index of the occurrence of `"c"` in `t`.

That is, the permutation difference between `s` and `t` is equal to `|0 - 1| + |1 - 0| + |2 - 2| = 2`.

Example 2:

Input: s = "abcde", t = "edbac"

Output: 12

Explanation: The permutation difference between `s` and `t` is equal to `|0 - 3| + |1 - 2| + |2 - 4| + |3 - 1| + |4 - 0| = 12`.

Constraints:

* `1 <= s.length <= 26` * Each character occurs at most once in `s`. * `t` is a permutation of `s`. * `s` consists only of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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