3146. Permutation Difference between Two Strings

Question

You are given two strings s and t such that:

1. Every character occurs at most once in s.
2. t is a permutation of s.

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

Return the permutation difference between s and t.

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.

Inputs

- s: A string of unique characters.
- t: A permutation of s.

Outputs

- An integer representing the permutation difference between s and t.

Example 1

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

Output:
2

Explanation:
The permutation difference is calculated as:

$|0-1|+|1-0|+|2-2|=2|0 - 1| + |1 - 0| + |2 - 2| = 2|0-1|+|1-0|+|2-2|=2$

Example 2

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

Output:
12

Explanation:
The permutation difference is calculated as:

$|0-3|+|1-2|+|2-4|+|3-1|+|4-0|=12|0 - 3| + |1 - 2| + |2 - 4| + |3 - 1| + |4 - 0| = 12|0-3|+|1-2|+|2-4|+|3-1|+|4-0|=12$

Algorithm

1. Initialize a variable perm_difference to 0.
2. Loop through each character in s using its index i.
    - Find the index of the same character in t using strchr().
    - Compute the absolute difference between the indices and add it to perm_difference.
3. Return perm_difference.

Code

```
#include <string.h>
#include <stdlib.h>

int findPermutationDifference(char* s, char* t) {
    int perm_difference = 0;
```

```
    for (int i = 0; i < strlen(s); i++) {
        // Find the index of s[i] in t
        int t_index = strchr(t, s[i]) - t;
        perm_difference += abs(t_index - i);
    }

    return perm_difference;
}
```

Time Complexity

- **$O(n^2)$:**
  - O(n) for looping through each character in s.
  - O(n) for each call to strchr(), which searches for a character in t.

Space Complexity

- **O(1):** No additional space is used beyond variables.

Edge Cases

1. **Minimal input size:**
   - Example: s = "a", t = "a" → Output: 0.
2. **Completely reversed strings:**
   - Example: s = "abc", t = "cba" → Output: $|0-2|+|1-1|+|2-0|=4$|0 - 2| + |1 - 1| + |2 - 0| = 4 $|0-2|+|1-1|+|2-0|=4$.
3. **Strings with maximum length (26):**
   - Ensure the algorithm handles up to 26 characters efficiently.