[CS 11 25.1] HOPE 1 C3 – Beep Boop

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Submit solution [CS 11 25.1] HOPE 1

Points: 230 (partial)
Time limit: 12.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Problem Statement

Is this all just a bad dream...?

All you can see in front of you is a string s consisting of lowercase letters, opening brackets (`[`), and closing brackets (`]`).

But...what is this string for? You see a play button below the string and decide to press it.

The string

```
a[ba[na]]q[ue]
```

suddenly turns into

```
abananabananaqueue
```

...that's odd.

You quickly realize that the brackets *double* the substring that they contain. If we look at this one bracket at a time, we get:

```
a[ba[na]]q[ue]

=> a[banana]q[ue]
       =-=-
=> abananabananaq[ue]
    =-----=-----
=> abananabananaqueue
                 =-=-
```

Let t be the string you obtain when you "expand" out all of the brackets in s. Count the number of times the character c appears among the characters between indices i and j (i is inclusive, j is **exclusive**).

Note that the string you obtain after expanding might be a bit long. Consider

```
[[[[[[[[[[ha]]]]]]]]]]
```

...would you like to try expanding it?

Task Details

Your task is to implement a function named `count_occ`. This function has four parameters: s, i, j, and c. s is a string, while the rest are integers.

The function must return an integer denoting the number of times c appears among the characters between indices i and j in the expansion of s.

Restrictions

- Your source code must have at most 6000 bytes.

Examples

Example 1 Function Call

```
count_occ("a[ba[na]]q[ue]", 1, 9, "a")
```

Example 1 Return Value

```
4
```

Example 2 Function Call

```
count_occ("ni[[ha]]", 0, 4, "k")
```

Example 2 Return Value

```
0
```

Constraints

- The function `count_occ` will be called at most 50 times.
- s consists of lowercase English letters, `[`, and `]`.
- The brackets in s are "matched properly"; that is, each `[` has a corresponding `]`.
- The length of s is at least 1 and at most 300.
- $0 \leq i < j \leq \text{len}(t)$
- $j \leq 10^{20}$
- c is a lowercase English letter.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 25 ❤️ points if you solve all test cases where:
  - There are no `[` or `]` in s.
- You get 75 ❤️ points if you solve all test cases where:
  - There are no nested brackets.
- You get 20 🔴 points if you solve all test cases where:
  - The length of s is at most 80.
  - $j - i \leq 80$
- You get 60 🔴 points if you solve all test cases where:
  - $j - i \leq 80$
- You get 25 💔 points if you solve all test cases where:
  - The length of s is at most 80.
- You get 25 💔 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.