5873. Maximize the Confusion of an Exam

My Submissions (/contest/biweekly-contest-62/problems/maximize-the-confusion-of-an-exam/submissions/)

Back to Contest (/contest/biweekly-contest-62/)

A teacher is writing a test with `n` true/false questions, with `'T'` denoting true and `'F'` denoting false. He wants to confuse the students by **maximizing** the number of **consecutive** questions with the **same** answer (multiple trues or multiple falses in a row).

You are given a string `answerKey`, where `answerKey[i]` is the original answer to the i^{th} question. In addition, you are given an integer `k`, the maximum number of times you may perform the following operation:

- Change the answer key for any question to `'T'` or `'F'` (i.e., set `answerKey[i]` to `'T'` or `'F'`).

Return *the **maximum** number of consecutive* `'T'`s or `'F'`s *in the answer key after performing the operation at most* `k` *times*.

User Accepted:	2
User Tried:	2
Total Accepted:	2
Total Submissions:	2
Difficulty:	Medium

Example 1:

```
Input: answerKey = "TTFF", k = 2
Output: 4
Explanation: We can replace both the 'F's with 'T's to make answerKey = "TTTT".
There are four consecutive 'T's.
```

Example 2:

```
Input: answerKey = "TFFT", k = 1
Output: 3
Explanation: We can replace the first 'T' with an 'F' to make answerKey = "FFFT".
Alternatively, we can replace the second 'T' with an 'F' to make answerKey = "TFFF".
In both cases, there are three consecutive 'F's.
```

Example 3:

```
Input: answerKey = "TTFTTFTT", k = 1
Output: 5
Explanation: We can replace the first 'F' to make answerKey = "TTTTTFTT"
Alternatively, we can replace the second 'F' to make answerKey = "TTFTTTTT".
In both cases, there are five consecutive 'T's.
```

Constraints:

- `n == answerKey.length`
- `1 <= n <= 5 * 10^4`
- `answerKey[i]` is either `'T'` or `'F'`
- `1 <= k <= n`

JavaScript

```
const mx = Math.max;
const maxConsecutiveAnswers = (s, k) => {
    let res1 = find(s, 'F', k);
    let res2 = find(s, 'T', k);
    return mx(res1, res2);
};

const find = (s, c, k) => {
    let n = s.length, cnt = 0, res = 0;
    for (let i = 0, j = i; i < n; i++) {
```

```
        for (; j < n; j++) {
            if (s[j] == c) {
            } else if (cnt < k) {
                cnt++;
            } else {
                break;
            }
        }
        res = mx(res, j - i);
        if (s[i] != c) cnt--;
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/564584934/) | More Details > (/submissions/detail/564584934/)

Share your acceptance!