5462. String Compression II

My Submissions (/contest/weekly-contest-199/problems/string-compression-ii/submissions/)

Back to Contest (/contest/weekly-contest-199/)

Run-length encoding (https://en.wikipedia.org/wiki/Run-length_encoding) is a string compression method that works by replacing consecutive identical characters (repeated 2 or more times) with the concatenation of the character and the number marking the count of the characters (length of the run). For example, to compress the string `"aabccc"` we replace `"aa"` by `"a2"` and replace `"ccc"` by `"c3"` . Thus the compressed string becomes `"a2bc3"` .

Notice that in this problem, we are not adding `'1'` after single characters.

User Accepted:	0
User Tried:	3
Total Accepted:	0
Total Submissions:	3
Difficulty:	Hard

Given a string `s` and an integer `k` . You need to delete **at most** `k` characters from `s` such that the run-length encoded version of `s` has minimum length.

Find the *minimum length of the run-length encoded version of* `s` *after deleting at most* `k` *characters*.

Example 1:

```
Input: s = "aaabcccd", k = 2
Output: 4
Explanation: Compressing s without deleting anything will give us "a3bc3d" of length 6. De
```

Example 2:

```
Input: s = "aabbaa", k = 2
Output: 2
Explanation: If we delete both 'b' characters, the resulting compressed string would be "a
```

Example 3:

```
Input: s = "aaaaaaaaaa", k = 0
Output: 3
Explanation: Since k is zero, we cannot delete anything. The compressed string is "a11" of
```

Constraints:

- `1 <= s.length <= 100`

- `0 <= k <= s.length`
- `s` contains only lowercase English letters.

JavaScript ▾

```
1 ▾ /**
2    * @param {string} s
3    * @param {number} k
4    * @return {number}
5    */
6 ▾ var getLengthOfOptimalCompression = function(s, k) {
7
8   };
```

☐ **Custom Testcase**   (Use Example Testcases)

▶ Run   ☁ Submit