6093. Design a Text Editor

My Submissions (/contest/weekly-contest-296/problems/design-a-text-editor/submissions/) Back to Contest (/contest/weekly-contest-296/)

Design a text editor with a cursor that can do the following:

- · Add text to where the cursor is.
- Delete text from where the cursor is (simulating the backspace key).
- . Move the cursor either left or right.

When deleting text, only characters to the left of the cursor will be deleted. The cursor will also remain within the actual text and cannot be moved beyond it. More formally, we have that 0 <= cursor.position <= currentText.length always holds.

User Accepted: 0

User Tried: 0

Total Accepted: 0

Total Submissions: 0

Difficulty: Hard

o (/problems

aueens-ii/

Implement the TextEditor class:

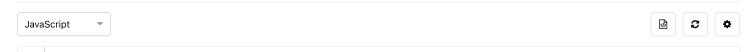
- TextEditor() Initializes the object with empty text.
- void addText(string text) Appends text to where the cursor is. The cursor ends to the right of text.
- int deleteText(int k) Deletes k characters to the left of the cursor. Returns the number of characters actually deleted.
- string cursorLeft(int k) Moves the cursor to the left k times. Returns the last min(10, len) characters to the left of the cursor, where len is the number of characters to the left of the cursor.
- string cursorRight(int k) Moves the cursor to the right k times. Returns the last min(10, len) characters to the left of the cursor, where len is the number of characters to the left of the cursor.

Example 1:

```
Input
["TextEditor", "addText", "deleteText", "addText", "cursorRight", "cursorLeft", "deleteText", "cursorLeft", "curso
[[], ["leetcode"], [4], ["practice"], [3], [8], [10], [2], [6]]
Output
[null, null, 4, null, "etpractice", "leet", 4, "", "practi"]
TextEditor textEditor = new TextEditor(); // The current text is "|". (The '|' character represents the cursor)
textEditor.addText("leetcode"); // The current text is "leetcode|".
textEditor.deleteText(4); // return 4
                                                           // The current text is "leet|".
                                                            // 4 characters were deleted.
textEditor.addText("practice"); // The current text is "leetpractice|".
textEditor.cursorRight(3); // return "etpractice"
                                                              // The current text is "leetpractice|".
                                                              // The cursor cannot be moved beyond the actual text and thus did not move.
                                                             // "etpractice" is the last 10 characters to the left of the cursor.
textEditor.cursorLeft(8); // return "leet"
                                                           // The current text is "leet|practice".
                                                           // "leet" is the last min(10, 4) = 4 characters to the left of the cursor.
textEditor.deleteText(10); // return 4
                                                              // The current text is "|practice".
                                                              // Only 4 characters were deleted.
textEditor.cursorLeft(2); // return ""
                                                           // The current text is "|practice".
                                                           // The cursor cannot be moved beyond the actual text and thus did not move.
                                                           // "" is the last min(10, 0) = 0 characters to the left of the cursor.
textEditor.cursorRight(6); // return "practi"
                                                              // The current text is "practi|ce".
                                                              // "practi" is the last min(10, 6) = 6 characters to the left of the cursor.
```

Constraints:

- 1 <= text.length, k <= 40
- text consists of lowercase English letters.
- At most $2 * 10^4$ calls in total will be made to addText, deleteText, cursorLeft and cursorRight.



```
1 ▼ function TextEditor() {
        let L = '', R = '
        return { addText, deleteText, cursorLeft, cursorRight }
 3
 4 •
        function addText(text) {
 5
           L += text;
 6
 7 ,
        function deleteText(k) {
            let remove = Math.min(k, L.length);
 8
 9
            L = L.slice(0, L.length - remove);
10
            return remove;
11
12 •
        function cursorLeft(k) {
13
            let leftmove = Math.min(L.length, k);
            R = L.slice(L.length - leftmove) + R;
14
            L = L.slice(0, L.length - leftmove);
15
16
            let t = Math.min(10, L.length);
17
            return L.slice(L.length - t);
18
19 ▼
        function cursorRight(k) {
20
            let leftmove = Math.min(R.length, k);
            L += R.slice(0, leftmove);
21
22
            R = R.slice(leftmove);
23
            let t = Math.min(10, L.length);
            return L.slice(L.length - t);
24
25
        }
26
    }
```

☐ Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/714823733/) **②**

More Details ➤ (/submissions/detail/714823733/)

Share your acceptance!

■10

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)

△ Submit

Run