←―(/) Explore(/explore/)   Problems(/problemset/all/)   Interview ^New Contest ^(/contest/)   Discuss(/discuss/)   Leetcode is hiring! Apply NOW. (https://leetcode.com/jobs/)   ☆ Premium (/subscribe?ref=nb_npl)   🔔   ⬦ 0   (/problems of-two-link

2296. 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.

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.

User Accepted:	2088
User Tried:	4024
Total Accepted:	2249
Total Submissions:	9713
Difficulty:	Hard

Example 1:

```
Input
["TextEditor", "addText", "deleteText", "addText", "cursorRight", "cursorLeft", "deleteText", "cursorLeft", "cursorRight"]
[[], ["leetcode"], [4], ["practice"], [3], [8], [10], [2], [6]]
Output
[null, null, 4, null, "etpractice", "leet", 4, "", "practi"]

Explanation
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`.

| Discuss (https://leetcode.com/problems/design-a-text-editor/discuss) |

JavaScript ▼

```javascript
function TextEditor() {
    let L = [], R = [];
    return { addText, deleteText, cursorLeft, cursorRight }
    function addText(text) {
        for (const c of text) L.push(c);
    }
    function deleteText(k) {
        let remove = Math.min(k, L.length);
        for (let i = 0; i < remove; i++) L.pop();
        return remove;
    }
    function cursorLeft(k) {
        let leftmove = Math.min(L.length, k);
        while (leftmove--) {
            let c = L.pop();
            R.push(c);
        }
        let t = Math.min(10, L.length);
        return L.slice(L.length - t).join("");
    }
    function cursorRight(k) {
        let leftmove = Math.min(R.length, k);
        while (leftmove--) {
            let c = R.pop();
            L.push(c);
        }
        let t = Math.min(10, L.length);
        return L.slice(L.length - t).join("");
    }
}
```

☐ **Custom Testcase**   Use Example Testcases

▶ Run   ☁ Submit

Submission Result: *Accepted* (/submissions/detail/715347461/) ❓   More Details ❯ (/submissions/detail/715347461/)

Share your acceptance!

◁ **12**