JavaScript Backspace String Compare

Challenge

Given two strings `s` and `t`, return `true` if they are equal when both are typed into empty text editors. `'#'` means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

1st Example

```
Input: s = 'ab#c', t = 'ad#c'
Output: true
Explanation: Both s and t become 'ac'.
```

2nd Example

```
Input: s = 'ab##', t = 'c#d#'
Output: true
Explanation: Both s and t become ''.
```

3rd Example

```
Input: s = 'a#c', t = 'b'
Output: false
Explanation: s becomes 'c' while t becomes 'b'.
```

Constraints

- `1 <= s.length, t.length <= 200`
- `s` and `t` only contain lowercase letters and `'#'` characters.

Solution

```
const backspaceCompare = (s, t) => {
    const strip = (str) => {
        const stack = [];

        for (const char of str) {
            if (char === '#') {
                stack.length > 0 && stack.pop();
            } else {
                stack.push(char);
            }
        }

        return stack.join('');
    }

    return strip(s) === strip(t);
};
```

Explanation

I've written a function called `backspaceCompare` that takes in two strings `s` and `t` as parameters. The function compares the two strings after applying a backspace operation to each of them.

Firstly, this function defines an inner function called `strip` that takes in a string `str` as a parameter. This function removes the characters that are preceded by a backspace symbol (`#`) from the string. It uses a stack data structure to keep track of the characters. If a character is not a backspace symbol, it is pushed onto the stack. If a backspace symbol is encountered, the last character is popped from the stack if the stack is not empty. Finally, the function returns the string obtained by joining the characters in the stack.

The `backspaceCompare` function then calls the `strip` function with the string `s` and `t` as arguments. It compares the two resulting stripped strings and returns true if they are equal, indicating that both strings are the same after applying the backspace operation. Otherwise, it returns false.

In summary, the `backspaceCompare` function compares two strings after applying a backspace operation to each of them. It uses an inner function called `strip` to remove characters preceded by a backspace symbol from a string. The function returns true if the stripped strings are equal, and false otherwise.

Author: Trevor Morin