⟨/⟩ Backspace String Compare

Solution 📄

Given two strings `S` and `T`, return 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.

Example 1:

```
Input: S = "ab#c", T = "ad#c"
Output: true
Explanation: Both S and T become "ac".
```

Example 2:

```
Input: S = "ab##", T = "c#d#"
Output: true
Explanation: Both S and T become "".
```

Example 3:

```
Input: S = "a##c", T = "#a#c"
Output: true
Explanation: Both S and T become "c".
```

Example 4:

```
Input: S = "a#c", T = "b"
Output: false
Explanation: S becomes "c" while T becomes "b".
```

Note:

- `1 <= S.length <= 200`
- `1 <= T.length <= 200`
- `S` and `T` only contain lowercase letters and `'#'` characters.

Follow up:

- Can you solve it in `O(N)` time and `O(1)` space?

Java ▾   📋 🔄 ⚙

```java
class Solution {
    public boolean backspaceCompare(String S, String T) {
        return buildString(S).equals(buildString(T));
    }

    private String buildString(String S) {
        StringBuilder str = new StringBuilder();
        for(char s : S.toCharArray()) {
            int length = str.length();
            if(s == '#') {
                if(length != 0) {
                    str.deleteCharAt(length-1);
                }
            }
            else {
                str.append(s);
            }
        }

        return str.toString();
    }

    /*private String buildString(String S) {
        Stack<Character> stack = new Stack();
        for(char s : S.toCharArray()) {
            if(s != '#') {
                stack.push(s);
            }
```

```
29            else {
30                if(!stack.isEmpty()) {
31                    stack.pop();
32                }
33            }
34        }
35
36        StringBuilder str = new StringBuilder();
37        while(!stack.isEmpty()) {
38            str.insert(0, stack.pop());
39        }
40
41        return str.toString();
42    }*/
43 }
```

Custom Testcase (**Contribute ❶**)

❓   ▶ Run Code   ☁ Submit