

6161. Removing Stars From a String

My Submissions (/contest/weekly-contest-308/problems/removing-stars-from-a-string/submissions/)

Back to Contest (/contest/weekly-contest-308/)

You are given a string `s` , which contains stars `*` .

In one operation, you can:

- Choose a star in `s` .
- Remove the closest **non-star** character to its **left**, as well as remove the star itself.

Return *the string after **all** stars have been removed*.

Note:

- The input will be generated such that the operation is always possible.
- It can be shown that the resulting string will always be unique.

User Accepted:	170
User Tried:	180
Total Accepted:	170
Total Submissions:	182
Difficulty:	Medium

Example 1:

Input: `s = "leet**cod*e"`

Output: `"lecoe"`

Explanation: Performing the removals from left to right:

- The closest character to the 1st star is 't' in `"leet**cod*e"`. `s` becomes `"lee*cod*e"`.
- The closest character to the 2nd star is 'e' in `"lee*cod*e"`. `s` becomes `"lecod*e"`.
- The closest character to the 3rd star is 'd' in `"lecod*e"`. `s` becomes `"lecoe"`.

There are no more stars, so we return `"lecoe"`.

Example 2:

Input: `s = "erase*****"`

Output: `""`

Explanation: The entire string is removed, so we return an empty string.

Constraints:

- `1 <= s.length <= 105`
- `s` consists of lowercase English letters and stars `*` .
- The operation above can be performed on `s` .

JavaScript

📄 ↺ ⚙️

```
1 const removeStars = (s) => {
2   let st = [];
3   for (const c of s) {
4     if (c == '*') {
5       st.pop();
6     } else {
7       st.push(c);
8     }
9   }
10  return st.join("");
11 };
```

☐ Custom Testcase

Use Example Testcases

Run

Submit

Submission Result: **Accepted** (/submissions/detail/785188763/) ⓘ More Details ▶ (/submissions/detail/785188763/)

Share your acceptance!

◀ 1