Problem 2390: Removing Stars From a String

Problem Information

Difficulty: Medium
Acceptance Rate: 78.63%
Paid Only: No
Tags: String, Stack, Simulation

Problem Description

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.

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 "lee** _t_** **cod*e". s becomes "lee*cod*e". - The closest character to the 2nd star is 'e' in "le** _e_** *cod*e". s becomes "lecod*e". - The closest character to the 3rd star is 'd' in "leco** _d_** *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`.

Code Snippets

C++:

```
class Solution {
public:
string removeStars(string s) {

}
};
```

Java:

```
class Solution {
public String removeStars(String s) {

}
}
```

Python3:

```
class Solution:
def removeStars(self, s: str) -> str:
```