

Description

Solution

Discuss (999+)

Submissions

## 1047. Remove All Adjacent Duplicates In String

Easy

2835

147

Add to List

Share

You are given a string `s` consisting of lowercase English letters. A **duplicate removal** consists of choosing two **adjacent** and **equal** letters and removing them.

We repeatedly make **duplicate removals** on `s` until we no longer can.

Return *the final string after all such duplicate removals have been made*. It can be proven that the answer is **unique**.

### Example 1:

**Input:** `s = "abbaca"`

**Output:** `"ca"`

**Explanation:**

For example, in "abbaca" we could remove "bb" since the letters are adjacent and equal, and this is the only possible move. The result of this move is that the string is "aaca", of which only "aa" is possible, so the final string is "ca".

### Example 2:

**Input:** `s = "azxxzy"`

**Output:** `"ay"`

### Constraints:

- `1 <= s.length <= 105`
- `s` consists of lowercase English letters.

Accepted 255,996

Submissions 360,478

Seen this question in a real interview before?

Yes

No

Companies

i

Java

Autocomplete

i

{ }

```
1  class Solution {
2      public String removeDuplicates( String s ) {
3
4          Stack<Character> stack = new Stack<>();
5
6          for (char ch : s.toCharArray()) {
7              if (!stack.isEmpty() && stack.peek() == ch) {
8                  stack.pop();
9              } else {
10                 stack.push(ch);
11             }
12         }
13
14         StringBuilder stringBuilder = new StringBuilder();
15
16         while (!stack.isEmpty()) {
17             stringBuilder.append(stack.pop());
18         }
19
20         return stringBuilder.reverse().toString();
21     }
22 }
```

Your previous code was restored from your local storage. [Reset to default](#)