

316. Remove Duplicate Letters

Given a string `s`, remove duplicate letters so that every letter appears once and only once. You must make sure your result is **the smallest in lexicographical order** among all possible results.

Example 1:

Input: `s = "bcabc"`

Output: `"abc"`

Example 2:

Input: `s = "cbacdcbc"`

Output: `"acdb"`

Constraints:

- `1 <= s.length <= 104`
- `s` consists of lowercase English letters.

Note: This question is the same as 1081: <https://leetcode.com/problems/smallest-subsequence-of-distinct-characters/>

```
class Solution:
    def removeDuplicateLetters(self, s: str) -> str:
        exists = [False] * 26
        freq = Counter(s)
        stack = []

        for i in range(len(s)):
            ch = s[i]
            freq[ch] = freq[ch] - 1
            if exists[ord(ch) - ord('a')]:
                continue
            while len(stack) and stack[-1] > ch and freq[stack[-1]] > 0:
                temp = stack.pop()
                exists[ord(temp) - ord('a')] = False
            stack.append(ch)
            exists[ord(ch) - ord('a')] = True
        return ''.join(stack)
```