

316 去除重复字符

Label: 栈

给你一个字符串 `s`，请你去除字符串中重复的字母，使得每个字母只出现一次。需保证 返回结果的字典序最小（要求不能打乱其他字符的相对位置）。

输入: `s = "bcabc"`

输出: `"abc"`

输入: `s = "cbacdcbc"`

输出: `"acdb"`

`1 <= s.length <= 104`

`s` 由小写英文字母组成

- 单调栈（直接用SB代替）

```
class Solution {
    public String removeDuplicateLetters(String s) {
        boolean[] vis = new boolean[26]; // 记录目前是否进串
        int[] num = new int[26];
        for (int i = 0; i < s.length(); i++) {
            num[s.charAt(i) - 'a']++; // 统计
        }
        StringBuffer sb = new StringBuffer();
        for (char ch : s.toCharArray()) {
            if (!vis[ch - 'a']) { // 没有遍历过
                while (sb.length() > 0 && sb.charAt(sb.length() - 1) > ch) { //
                    // 如果最后一个字符 顺序在当前字符之后
                    if (num[sb.charAt(sb.length() - 1) - 'a'] > 0) { // 如果最后
                        // 一个字符出现大于1次，那么说明可以出栈，后续再加入
                        vis[sb.charAt(sb.length() - 1) - 'a'] = false;
                        sb.deleteCharAt(sb.length() - 1); // 出栈
                    } else {
                        break;
                    }
                }
                vis[ch - 'a'] = true;
                sb.append(ch); // 遍历过
            }
            num[ch - 'a']--;
        }
        return sb.toString();
    }
}
```

- 单调栈 (用栈)

```
class Solution {
    public String removeDuplicateLetters(String s) {
        Stack<Character> stk = new Stack<>();
        int[] count = new int[26];
        for (int i = 0; i < s.length(); i++) {
            count[s.charAt(i) - 'a']++; // 统计
        }
        boolean[] inStack = new boolean[26];

        for (char c : s.toCharArray()) {
            count[c - 'a']--;
            if (inStack[c - 'a']) continue;

            while (!stk.isEmpty() && stk.peek() > c) {
                // 若之后不存在栈顶元素了, 则停止 pop
                if (count[stk.peek() - 'a'] == 0) {
                    break;
                }
                // 若之后还有, 则可以 pop
                inStack[stk.pop() - 'a'] = false;
            }
            stk.push(c);
            inStack[c - 'a'] = true;
        }

        StringBuilder sb = new StringBuilder();
        while (!stk.empty()) {
            sb.append(stk.pop());
        }
        return sb.reverse().toString();
    }
}
```