5月29日日报

本日学习内容

- 1. 继续学习ui 学习ui控制器和工具栏
- 2. 继续完成UI博客

今日算法题

题目1: 394. 字符串解码

394. 字符串解码 □解答 ⊙



给定一个经过编码的字符串,返回它解码后的字符串。

编码规则为: k[encoded_string] ,表示其中方括号内部的 encoded_string 正好重复 k 次。注意 k 保证为正整数。

你可以认为输入字符串总是有效的;输入字符串中没有额外的空格,且输入的方括号总是符合格式要求的。

此外,你可以认为原始数据不包含数字,所有的数字只表示重复的次数 k ,例如不会出现像 3a 或 2[4] 的输入。

示例 1:

```
输出: "aaabcbc"
示例 2:
输入: s = "3[a2[c]]"
输出: "accaccacc"
示例 3:
```

输入: s = "3[a]2[bc]"

输入: s = "2[abc]3[cd]ef" 输出: "abcabccdcdcdef"

示例 4:

输入: s = "abc3[cd]xyz" 输出: "abccdcdcdxyz"

提示:

- 1 <= s.length <= 30
- s 由小写英文字母、数字和方括号 '[]' 组成
- 写 保证是一个 有效 的输入。
- s 中所有整数的取值范围为 [1, 300]

```
class Solution {
public:
    string decodeString(string s) {
    stack<int> countStack;
```

```
stack<string> stringStack;
    string currentString = "";
    int currentNum = 0;
    for (char ch : s) {
        if (isdigit(ch)) {
            currentNum = currentNum * 10 + (ch - '0');
        } else if (ch == '[') {
            countStack.push(currentNum);
            stringStack.push(currentString);
            currentString = "";
            currentNum = 0;
        } else if (ch == ']') {
            string prevString = stringStack.top(); stringStack.pop();
            int repeatTimes = countStack.top(); countStack.pop();
            string temp = "";
            for (int i = 0; i < repeatTimes; ++i) {</pre>
                temp += currentString;
            currentString = prevString + temp;
        } else {
            currentString += ch;
    }
    return currentString;
    }
};
```

本日遇到的问题

1. UI部分内容还不能完全掌握,写程序时还需要看之前写过的程序或笔记

明日学习计划

1. 复习之前学过的控件,继续向后深入学习控制器和工具栏