

5月28日报

本日学习内容

1. 继续学习ui UITextField, UIScrollView, 步进器与分栏控件, UISwitch
2. 继续完成UI博客

今日算法题

题目1: [139. 单词拆分](#)

139. 单词拆分

已解答 

中等

 相关标签

 相关企业

Aa

给你一个字符串 `s` 和一个字符串列表 `wordDict` 作为字典。如果可以利用字典中出现的一个或多个单词拼接出 `s` 则返回 `true`。

注意：不要求字典中出现的单词全部都使用，并且字典中的单词可以重复使用。

示例 1:

输入: `s = "leetcode", wordDict = ["leet", "code"]`
输出: `true`
解释: 返回 `true` 因为 "leetcode" 可以由 "leet" 和 "code" 拼接成。

示例 2:

输入: `s = "applepenapple", wordDict = ["apple", "pen"]`
输出: `true`
解释: 返回 `true` 因为 "applepenapple" 可以由 "apple" "pen" "apple" 拼接成。
注意，你可以重复使用字典中的单词。

示例 3:

输入: `s = "catsanddog", wordDict = ["cats", "dog", "sand", "and", "cat"]`
输出: `false`

提示:

- `1 <= s.length <= 300`
- `1 <= wordDict.length <= 1000`
- `1 <= wordDict[i].length <= 20`
- `s` 和 `wordDict[i]` 仅由小写英文字母组成
- `wordDict` 中的所有字符串 互不相同

```
class Solution {
public:
    bool wordBreak(string s, vector<string>& wordDict) {
        unordered_set<string> dict(wordDict.begin(), wordDict.end());
        int n = s.size();
```

```
vector<bool> dp(n + 1, false);
dp[0] = true; // 空串可以拆分

for (int i = 1; i <= n; ++i) {
    for (int j = 0; j < i; ++j) {
        if (dp[j] && dict.count(s.substr(j, i - j))) {
            dp[i] = true;
            break; // 找到一种拆分方式就可以停止了
        }
    }
}

return dp[n];
}
```

本日遇到的问题

1. UI部分控件较多，还不能完全掌握，写程序时还需要看笔记

明日学习计划

1. 复习之前学过的控件，完成登录系统