

0140. 单词拆分 II

👤 ITCharge ⌚ 大约 1 分钟

- 标签：字典树、记忆化搜索、数组、哈希表、字符串、动态规划、回溯
- 难度：困难

题目链接

- [0140. 单词拆分 II - 力扣](#)

题目大意

给定一个非空字符串 `s` 和一个包含非空单词列表的字典 `wordDict`。

要求：在字符串中增加空格来构建一个句子，使得句子中所有的单词都在词典中。返回所有这些可能的句子。

说明：

- 分隔时可以重复使用字典中的单词。
- 你可以假设字典中没有重复的单词。

解题思路

回溯 + 记忆化搜索。

对于字符串 `s`，如果某个位置左侧部分是单词列表中的单词，则拆分出该单词，然后对 `s` 右侧剩余部分进行递归拆分。如果可以将整个字符串 `s` 拆分成单词列表中的单词，则得到一个句子。

使用 `memo` 数组进行记忆化存储，这样可以减少重复计算。

代码

py

```
class Solution:
    def wordBreak(self, s: str, wordDict: List[str]) -> List[str]:
        size = len(s)
        memo = [None for _ in range(size + 1)]

        def dfs(start):
            if start > size - 1:
                return [[]]
            if memo[start]:
                return memo[start]
            res = []
            for i in range(start, size):
                word = s[start: i + 1]
                if word in wordDict:
                    rest_res = dfs(i + 1)
                    for item in rest_res:
                        res.append(word + item)
            memo[start] = res
            return res

        res = dfs(0)
        ans = []
        for item in res:
            ans.append(" ".join(item))
        return ans
```