140-单词拆分II

题述

140. 单词拆分 II

难度 困难 596

给定一个字符串 `s` 和一个字符串字典 `wordDict` ，在字符串 `s` 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。**以任意顺序** 返回所有这些可能的句子。

注意：词典中的同一个单词可能在分段中被重复使用多次。

示例 1：

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

示例 2：

```
输入:s = "pineapplepenapple", wordDict = ["apple","pen","applepen","pine","pineapple"]
输出:["pine apple pen apple","pineapple pen apple","pine applepen apple"]
解释: 注意你可以重复使用字典中的单词。
```

示例 3：

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

思路

使用回溯法的模板去进行编码

看代码去理解

题解

这个版本会超时

```python
class Solution:
    def wordBreak(self, s: str, wordDict: List[str]) -> List[str]:
        result = [] #结果数组
        wordDict = set(wordDict)    #使用set进行去重

        def BackTracking(wordDict,temp,pos):
```

```
            #递归函数
            if pos == len(s):    #pos为遍历到的位置 类似于startIndex
                #递归终止条件
                result.append(" ".join(temp))
                return
            for i in range(pos,len(s)+1):    #横向遍历
                if s[pos:i] in wordDict:
                    #判断是不是在单词字典中 使用in
                    temp.append(s[pos:i])
                    BackTracking(wordDict,temp,i)
                    temp.pop()
        BackTracking(wordDict,[],0)
        return result
```

剪枝优化

```
class Solution:
    # 带备忘录的记忆化搜索
    def wordBreak(self, s: str, wordDict: List[str]) -> List[str]:
        result = []
        memo = [1] * (len(s)+1)
        wordDict = set(wordDict)

        def BackTracking(wordDict,temp,pos):
            num = len(res)                    # 回溯前先记下答案中有多少个元素
            if pos == len(s):
                res.append(" ".join(temp))
                return
            for i in range(pos,len(s)+1):
                if memo[i] and s[pos:i] in wordDict: # 添加备忘录的判断条件
                    temp.append(s[pos:i])
                    dfs(wordDict,temp,i)
                    temp.pop()
            # 答案中的元素没有增加，说明s[pos:]不能分割，修改备忘录
            memo[pos] = 1 if len(res) > num else 0

        BackTracking(wordDict,[],0)
        return result
```

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	40 ms	15 MB	Python3	2022/05/26 10:31	添加备注
执行出错	N/A	N/A	Python3	2022/05/26 10:31	添加备注
执行出错	N/A	N/A	Python3	2022/05/26 10:30	添加备注
通过	40 ms	14.9 MB	Python3	2022/05/26 10:25	添加备注

思考