

29

9

# 139. 单词拆分 📖

Notes

题目描述 (/problems/word-break/description/)

提示帮助 (/problems/word-break/hints/)

提交记录 (/problems/word-break/submissions/)

随机一题 (/problems/random-one-question/)

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*，判定 *s* 是否可以被空格拆分为一个或多个在字典中出现的单词。

说明：

- 拆分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。

示例 1：

输入：s = "leetcode", wordDict = ["leet", "code"]  
输出：true  
解释：返回 true 因为 "leetcode" 可以被拆分成 "leet code"。

示例 2：

输入：s = "applepenapple", wordDict = ["apple", "pen"]  
输出：true  
解释：返回 true 因为 "applepenapple" 可以被拆分成 "apple pen apple"。  
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

您是否在真实的面试环节中遇到过这道题目呢？ 

是的

没有

✖

相关话题 ▾


相似题目 ▾


C++ ▾

```

2 class Solution {
3 public:
4     bool wordBreak(string s, vector<string>& wordDict) {
5         int n = s.length();
6         if(n==0)return true;
7         int max = wordDict.size();
8         if(max==0)return false;
9         int temp, i, j, index;
10        bool dp[n];
11        for(i=0;i<n;i++){
12            dp[i] = false;
13            for(i=0;i<max;i++){
14                temp = s.find(wordDict[i]);
15                index = wordDict[i].length()-1;
16                if(temp==0){
17                    // 第一个位置的元素被找到
18                    dp[index]=true;
19                }
20            }
21            for(int i=0;i<n-1;i++){
22                if(dp[i]){
23                    // 如果到这里是可以的
24                    for(j=0;j<max;j++){
25                        temp = s.find(wordDict[j], i+1);
26                        if(temp==i+1){
27                            index = wordDict[j].length()+temp-1;
28                            if(index==n-1)return true;
29                            dp[index] = true;
30                        }
31                    }
32                }
33                // 如果为假, 不处理
34            }
35            return dp[n-1];
36        }
37    };

```

☐ 自定义测试用例（贡献给我们 )

 执行代码

 提交解答

提交结果: **通过** (/submissions/detail/4748857/) 

更多明细  (/submissions/detail/4748857/)

进行下一个挑战: **单词拆分 II** (/problems/word-break-ii)

炫耀一下结果



