

Description

Solution

Discuss (999+)

Submissions

139. Word Break

Medium

4642

240

Add to List

Share

Given a **non-empty** string *s* and a dictionary *wordDict* containing a list of **non-empty** words, determine if *s* can be segmented into a space-separated sequence of one or more dictionary words.

- Note:**
- The same word in the dictionary may be reused multiple times in the segmentation.
 - You may assume the dictionary does not contain duplicate words.

Example 1:

Input: `s = "leetcode", wordDict = ["leet", "code"]`

Output: `true`

Explanation: Return true because "leetcode" can be segmented as "leet code".

Example 2:

Input: `s = "applepenapple", wordDict = ["apple", "pen"]`

Output: `true`

Explanation: Return true because "applepenapple" can be segmented as "apple pen apple".

Note that you are allowed to reuse a dictionary word.

Example 3:

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

Output: `false`

Accepted 575,044

Submissions 1,434,422

Java

Autocomplete

```
1 class Solution {
2     public boolean wordBreak(String s, List<String> wordList) {
3         boolean[] T = new boolean[s.length() + 1];
4         Set<String> set = new HashSet<>();
5         for (String word : wordList) {
6             set.add(word);
7         }
8         T[0] = true;
9         for (int i = 1; i <= s.length(); i++) {
10            for (int j = 0; j < i; j++) {
11                if(T[j] && set.contains(s.substring(j, i))) {
12                    T[i] = true;
13                    break;
14                }
15            }
16        }
17        return T[s.length()];
18    }
19 }
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

"leetcode"

["leet", "code"]

Output

true

Diff

Expected

true