

**179. Given a string `s` and a dictionary of strings `wordDict`, return true if `s` can be segmented into a space-separated sequence of one or more dictionary words.**

**Note that the same word in the dictionary may be reused multiple times in the segmentation.**

**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 = "catsanddog", wordDict = ["cats", "dog", "sand", "and", "cat"]`

**Output:** false

**Program:**

```
def word_break(s, wordDict):
```

```
    word_set = set(wordDict)
```

```
    dp = [False] * (len(s) + 1)
```

```
    dp[0] = True
```

```
    for i in range(1, len(s) + 1):
```

```
        for j in range(i):
```

```
            if dp[j] and s[j:i] in word_set:
```

```
                dp[i] = True
```

```
                break
```

```
    return dp[len(s)]
```

**# Example 1**

**s1 = "leetcode"**

```
wordDict1 = ["leet", "code"]  
print(word_break(s1, wordDict1)) # Output: True
```

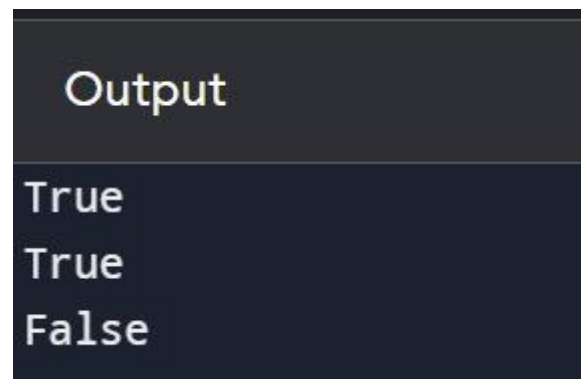
# Example 2

```
s2 = "applepenapple"  
wordDict2 = ["apple", "pen"]  
print(word_break(s2, wordDict2)) # Output: True
```

# Example 3

```
s3 = "catsandog"  
wordDict3 = ["cats", "dog", "sand", "and", "cat"]  
print(word_break(s3, wordDict3)) # Output: False
```

**Output:**



```
Output  
True  
True  
False
```

**Timecomplexity:  $O(n^3)$**