Problem 139: Word Break

Problem Information

Difficulty: Medium
Acceptance Rate: 48.83%
Paid Only: No
Tags: Array, Hash Table, String, Dynamic Programming, Trie, Memoization

Problem Description

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 = "catsandog", wordDict = ["cats","dog","sand","and","cat"] **Output:** false

Constraints:

* `1 <= s.length <= 300` * `1 <= wordDict.length <= 1000` * `1 <= wordDict[i].length <= 20` * `s` and `wordDict[i]` consist of only lowercase English letters. * All the strings of `wordDict` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
bool wordBreak(string s, vector<string>& wordDict) {

}
};
```

Java:

```
class Solution {
public boolean wordBreak(String s, List<String> wordDict) {

}
}
```

Python3:

```
class Solution:
def wordBreak(self, s: str, wordDict: List[str]) -> bool:
```