139 单词拆分

Label：动态规划
给定一个非空字符串 s 和一个包含非空单词的列表 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。
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"。
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。
输入: s = "catsandog", wordDict = ["cats", "dog", "sand", "and", "cat"]
输出: false

- 动态规划

```
class Solution {
    public boolean wordBreak(String s, List<String> wordDict) {
       // 可以类比于背包问题
        int n = s.length();
        // memo[i] 表示 s 中以 i - 1 结尾的字符串是否可被 wordDict 拆分
        boolean[] memo = new boolean[n + 1];
        memo[0] = true;
        for (int i = 1; i <= n; i++) {
            for (int j = 0; j < i; j++) {
                 // 前j-1个元素组成的字符串可以被拆分，并且j到 i-1组成的字符串也在，则前i-1个元素可以被拆分
                if (memo[j] && wordDict.contains(s.substring(j, i))) {
                    memo[i] = true;  // memo[i] = memo[j] && wordDict.contains(s.substring(j, i)) 状态转移方程
                    break;
                }
            }
        }
        return memo[n];
    }
}
```