

Problem 140: Word Break II

Problem Information

Difficulty: Hard

Acceptance Rate: 54.61%

Paid Only: No

Tags: Array, Hash Table, String, Dynamic Programming, Backtracking, Trie, Memoization

Problem Description

Given a string `s` and a dictionary of strings `wordDict`, add spaces in `s` to construct a sentence where each word is a valid dictionary word. Return all such possible sentences in **any order**.

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

Example 1.

Input: `s = "catsanddog", wordDict = ["cat", "cats", "and", "sand", "dog"]` **Output:** `["cats and dog", "cat sand dog"]`

Example 2.

Input: `s = "pineapplepenapple", wordDict = ["apple", "pen", "applepen", "pine", "pineapple"]`
Output: `["pine apple pen apple", "pineapple pen apple", "pine applepen apple"]`
Explanation: Note that you are allowed to reuse a dictionary word.

Example 3.

Input: `s = "catsanddog", wordDict = ["cats", "dog", "sand", "and", "cat"]` **Output:** `[]`

Constraints:

`1 <= s.length <= 20` `1 <= wordDict.length <= 1000` `1 <= wordDict[i].length <= 10` `s` and `wordDict[i]` consist of only lowercase English letters. All the strings of `wordDict` are

****unique**.** * Input is generated in a way that the length of the answer doesn't exceed 105.

Code Snippets

C++:

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

    }
};
```

Java:

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

    }
}
```

Python3:

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