Word Break II

Problem Statement

Given a string s and a dictionary of strings wordDict, you need to add spaces in s to construct sentences 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 = "catsandog", 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 10^5.

Approach

To solve this problem, you can use dynamic programming to build a list of sentences that can be formed by breaking the string s into valid words from wordDict.

1. **Initialization:** Create a list dp where dp[i] holds all possible sentences that can be formed from the substring s[0:i]. Initialize dp[0] with an empty string (representing the base case where an empty string can be segmented in one way).

2. **Filling the DP Table:** Iterate through the string s using two nested loops:

- The outer loop iterates through each index i from 1 to len(s).
- The inner loop checks substrings s[j:i] for all possible starting indices j (from 0 to i-1). If s[j:i] is in the dictionary and dp[j] is not empty, append the valid substring to the sentences in dp[j].

3. **Returning the Result:** Return dp[len(s)], which contains all possible sentences that can be formed from the entire string s.

By following this approach, you can efficiently construct all valid sentences from the given string s and dictionary wordDict.