140. Word Break II

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

```
class Solution:
    def wordBreak(self, s: str, wordDict: List[str]) -> List[str]:
        res = []
        self.wordbreakUtil(s,wordDict,res,[])
        return res





    def wordbreakUtil(self,s,wordDict,res,ssf):
        if len(s)==0:
            temp = ' '.join(ssf)
            res.append(temp)
            return
        for i in range(len(s)):
            ss = s[0:i+1]
            if ss in wordDict:
                rest = s[i+1:]
                self.wordbreakUtil(rest,wordDict,res,ssf+[ss])
```