139. Word Break

Medium

1857107FavoriteShare

Given a non-empty string *s* and a dictionary *wordDict* containing a list of non-empty words, determine if *s* can be segmented into a space-separated sequence of one or more dictionary words.

Note:

* The same word in the dictionary may be reused multiple times in the segmentation.
* You may assume the dictionary does not contain duplicate words.

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

class Solution {

public:

bool wordBreak(string s, vector<string>& wordDict) {

if(wordDict.size()==0) return false;

vector<bool> dp(s.length(),false);

dp[0]=true;

for(int i=1;i<=s.size();i++){

for(int j=i-1;j>=0;j--){

if(dp[j]){

if(std::find(wordDict.begin(),wordDict.end(),s.substr(j,i-j))!=wordDict.end()){

dp[i]=true;

break;

}

}

}

}

return dp[s.size()];

}

};

Success

[Details](https://leetcode.com/submissions/detail/211375373/)

Runtime: 8 ms, faster than 99.55% of C++ online submissions for Word Break.

Memory Usage: 10.9 MB, less than 75.05% of C++ online submissions for Word Break.