140. Word Break II

My Submissions

QuestionEditorial Solution

Total Accepted: **59166** Total Submissions: **291290** Difficulty: **Hard**

Given a string *s* and a dictionary of words *dict*, add spaces in *s* to construct a sentence where each word is a valid dictionary word.

Return all such possible sentences.

For example, given

s = "catsanddog",

dict = ["cat", "cats", "and", "sand", "dog"].

A solution is ["cats and dog", "cat sand dog"].

Subscribe to see which companies asked this question

```
//author:ZZW
//C++
class Solution {
public:

void dfs(vector<bool> &space_loc,int loc,string si,string s,vector<string>
&ret,unordered_set<string>& wordDict)
    {
        if(loc==0)
        {
            ret.push_back(si);
        }else
        {
            for(int i=0;i<loc;i++)
            {
                if(space_loc[i] && wordDict.find(s.substr(i,loc-i))!=wordDict.end())
                {
                    dfs(space_loc,i,s.substr(i,loc-i)+ " "+si,s,ret,wordDict);
                }
            }
        }
    }

vector<string> wordBreak(string s, unordered_set<string>& wordDict) {
        vector<string> ret;
        int n = s.size();
        vector<bool> space_loc(n+1,false);
        space_loc[0] = true;
        // compute the max length
        int max_length = -1;
        unordered_set<string>::iterator it;
        for(it = wordDict.begin();it!=wordDict.end();it++)
        {
            int templength = it->length();
            if(templength>max_length)
                max_length = templength;
        }
        for(int i=0;i<n;i++)
        {
            if(space_loc[i])
            {
                for(int j=i+1;j<n+1 && j-i<=max_length;j++)
                {
                    if(wordDict.find(s.substr(i,j-i)) != wordDict.end())
                    {
                        space_loc[j]=true;
                    }
                }
                if(space_loc[i] && wordDict.find(s.substr(i,n-i))!=wordDict.end())
                {
                    dfs(space_loc,i,s.substr(i,n-i),s,ret,wordDict);
                }
            }
        }
        return ret;
    }
};
```