Word Break II

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"]`.

Solution 1

```
public class Solution {
public List<String> wordBreak(String s, Set<String> dict) {
    List<String> result = new ArrayList<String>();
    for(int j = s.length() - 1; j >= 0; j--){
        if(dict.contains(s.substring(j)))
            break;
        else{
            if(j == 0)
                return result;
        }
    }
    for(int i = 0; i < s.length()-1; i++)
    {
        if(dict.contains(s.substring(0,i+1)))
        {
            List<String> strs = wordBreak(s.substring(i+1,s.length()),dict);
            if(strs.size() != 0)
                for(Iterator<String> it = strs.iterator();it.hasNext();)
                {
                    result.add(s.substring(0,i+1)+" "+it.next());
                }
        }
    }
    if(dict.contains(s)) result.add(s);
    return result;
}
```

}

written by XingLiu original link here

Solution 2

```
class Solution {
    unordered_map<string, vector<string>> m;

    vector<string> combine(string word, vector<string> prev){
        for(int i=0;i<prev.size();++i){
            prev[i]+=" "+word;
        }
        return prev;
    }

public:
    vector<string> wordBreak(string s, unordered_set<string>& dict) {
        if(m.count(s)) return m[s]; //take from memory
        vector<string> result;
        if(dict.count(s)){ //a whole string is a word
            result.push_back(s);
        }
        for(int i=1;i<s.size();++i){
            string word=s.substr(i);
            if(dict.count(word)){
                string rem=s.substr(0,i);
                vector<string> prev=combine(word,wordBreak(rem,dict));
                result.insert(result.end(),prev.begin(), prev.end());
            }
        }
        m[s]=result; //memorize
        return result;
    }
};
```

written by samoshka original link here

Solution 3

firstly I used DP from head of the string to traverse the dp-map: and then got a “Time Limit Exceeded” Error with the unpassed case "aaaaaaaaa....ab", but this method can pass such case like "baaaaaa....a"

secondly I found the answer on internet with the dp-strategy, and saw the dp-method from tail of the string to traverse the dp-map, then got an "Accepted" ,but i tested the case like "baaaaaa....a" on my own computer ,finally the result is “Time Limit Exceeded”

above all, i think the two strategies are the same ; and the OJ's test cases may have some infulence on different methods!

written by CodingGod original link here