

一、题目说明

题目139. Word Break, 判断一个非空字符串是否能由字典中的字符串拼接成。难度是Medium!

二、我的解答

尽管这个题目难度是Medium, 遗憾的是没做出来。

用回溯方法, 从字符串s的第0个位置开始, ...如果s[0]在dict中出现, 继续查找。回溯法会超时的, 这里用visited数组表示是否使用过。

```
class Solution{
public:
    // 回溯法, 用visited裁剪 回溯的对象是s[0,1,...]
    bool wordBreak(string s, vector<string>& wordDict){
        if(s.empty()) return true;
        int n = s.size();
        //vector[i]表示从i下标开始的字符串的访问状态,0 表示未访问过,1 访问过但不可拆分
        // 2 访问过且可拆分
        vector<int> visited(n,0);
        if(dfs(s,wordDict,visited,0)){
            return true;
        }
        return false;
    }
    bool dfs(string s,vector<string>& wordDict,vector<int>& visited,int
begin){
        if(begin == s.size()) return true;
        for(int i=1;i<=s.size();i++){
            string temp = s.substr(begin,i);
            // 若已经访问过 判是否可达
            if(visited[begin] !=0){
                return visited[i]==2;
            }
            // 未访问过
            for(auto word:wordDict){
                if(word.compare(temp)==0){
                    if(dfs(s,wordDict,visited,begin+i)){
                        visited[begin] = 2;
                        return true;
                    }
                }
            }
        }

        visited[begin] = 1;
        return false;
    }
};
```

性能如下:

Runtime: 80 ms, faster than 5.58% of C++ online submissions for Word Break.
Memory Usage: 20.1 MB, less than 5.66% of C++ online submissions for Word Break.

三、优化措施

这个题目，用dp方法非常合适：

```
class Solution{
public:
    //dp solution:dp[i] means s[0..i] 能否有字典组成
    //dp[0] means "" can be made by dict
    bool wordBreak(string s, vector<string>& wordDict){
        vector<bool> dp(s.size()+1,false);
        dp[0] = true;
        for(int i=1;i<=s.size();i++){
            for(auto word: wordDict){
                int ws = word.size();
                if(i>=ws){
                    int cur = s.compare(i-ws,ws,word);
                    if(dp[i-ws] && cur==0){
                        dp[i] = true;
                    }
                }
            }
        }
        return dp[s.size()];
    }
};
```

优化措施：

Runtime: 8 ms, faster than 77.10% of C++ online submissions for Word Break.
Memory Usage: 8.8 MB, less than 94.34% of C++ online submissions for Word Break.