

139. Word Break

题目描述: <https://leetcode.com/problems/word-break/>

给定一个字符串，给定一个dict，如果该字符串能拆成该dict包括的串就return true
例如：

```
s = "leetcode",
dict = ["leet", "code"].
Return true because "leetcode" can be segmented as "leet code".
```

解题思路1:

$f[i][j]$ 代表从i到j（不包括j）能不能分割

$f[i][j] = (f[i][i+1] \&\& f[i+1][j]) \parallel (f[i][i+2] \&\& f[i+2][j]) \parallel \dots$

解题思路2:

$f[i]$ 代表以i结尾（不包括i）的字符串能不能组成 $f[i] = f[0] \&\& \text{word}[0][i] == \text{true} \parallel f[1] \&\& \text{word}[1][i] == \text{true} \parallel \dots$

代码1:

```
class Solution {
public:
    bool wordBreak(string s, vector<string>& wordDict) {
        set<string> st(wordDict.begin(), wordDict.end());
        int len = s.size();
        vector<vector<bool>> > f(len, vector<bool>(len+1, false));
        for(int l = 1; l <= len; l++) {
            for(int i = 0; i+l <= len; i++) {
                int j = i+l;
                if(st.find(s.substr(i, l)) != st.end()) {
                    f[i][j] = true;
                    continue;
                }
                for(int k = i+1; k < j; k++) {
                    if(f[i][k] &\& f[k][j]) {
                        f[i][j] = true;
                        continue;
                    }
                }
            }
        }
        return f[0][len];
    }
};
```

代码2:

```

class Solution {
public:
    bool wordBreak(string s, vector<string>& wordDict) {
        int len = s.size();
        set<string> st(wordDict.begin(), wordDict.end());
        vector<bool> f(len+1, false);
        f[0] = true;
        for(int i = 1; i <= len; i++) {
            for(int j = i-1; j >= 0; j--) {
                if(f[j]) {
                    string w = s.substr(j, i-j);
                    if(st.find(w) != st.end()) {
                        f[i] = true;
                        break;
                    }
                }
            }
        }
        return f[len];
    }
};

```