```java
class Solution {
    public boolean wordBreak(String s, List<String> wordDict) {
        Set<String> set = new HashSet<>();
        set.addAll(wordDict);
        int maxLen = getMaxLen(set);
        boolean[] bools = new boolean[s.length()+1];
        bools[0] = true;

        for (int i = 1; i <= s.length(); i++) {
            for (int j = i - 1; j >= 0 && j >= i - maxLen; j--) {
                if (bools[j]) {
                    String tmp = s.substring(j, i);
                    if (wordDict.contains(tmp)) {
                        bools[i] = true;
                    }
                }
            }
        }
        return bools[s.length()];
    }

    private int getMaxLen(Set<String> set) {
        int len = 0;
        for (String s : set) {
            len = Math.max(len, s.length());
        }
        return len;
    }

    /*
    public boolean wordBreak(String s, List<String> wordDict) {
        return dfs(s, 0, wordDict);
    }

    private boolean dfs(String s, int idx, List<String> wordDict) {
        if (idx == s.length())
            return true;
        String sub = s.substring(idx);
        for (String word : wordDict) {
            if (sub.startsWith(word) && dfs(s, idx+word.length(), wordDict)) {
                return true;
            }
        }
        return false;
    }
    */
}
```