**[Word Break](https://leetcode.com/problems/word-break/)**

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** WordBreak {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

List<String> wordDict = **new** ArrayList<>();

wordDict.add("leet");

wordDict.add("code");

System.***out***.println(*wordBreak*("leetcode" , wordDict));

}

**public** **static** **boolean** wordBreak(String s, List<String> wordDict) {

**boolean**[] isWord = **new** **boolean**[s.length() + 1];

isWord[0] = **true**;

**for**(**int** i = 0 ; i <= s.length() ; i++) {

**for**(**int** j = 0 ; j <= i ; j++) {

**if**(!isWord[j]) {

**continue**;

}

**if**(wordDict.contains(s.subSequence(j, i))) {

isWord[i] = **true**;

**break**;

}

}

}

**return** isWord[s.length()];

}

}

Time complexity : O(n) , n is length of given word

Space Complexity : O(n) , n is length of given word