Word Break

Given an input string and a dictionary of words, find out if the input string can be segmented into a space-separated sequence of dictionary words.

For example, consider the following dictionary: { pear, salmon, foot, prints, footprints, leave, you, sun, girl, enjoy },

Examples:

Given the string “youenjoy”,

Output: True (The string can be segmented as “you enjoy”)

Input: “youleavefootprints”,

Output: True (The string can be segmented as “you leave footprints” or “you leave foot prints”)

Input:salmonenjoyapples

Output: False

**T- Talk (/Listen/Clarify)**

**E-Examples (/Test/TDD, Out of the Box)**

dictionary: { pear, salmon, foot, prints, footprints, leave, you, sun, girl, enjoy }

|  |  |  |
| --- | --- | --- |
| Case | Input | Output |
| Empty dictionary | “pear” | False |
| Empty string | “” | False |
| Null | Null | Throws exception |
| Input tring that has repeated words | “girlenjoysalmonenjoysun” | True |
| Capitalized in input string but not in dictionary and vice versa | “youLeave” | True |
| Input string contains no words that are in the dictionary | “boblikescheese” | False |
| Input string contains some words that are in the dictionary | “sheleavesfootprints”  “footprintssheleaves” | False  False |
| All words in the input string are in the dictionary | “youenjoysalmon” | True |
| Input string contains non alphabetic characters | “123@salmon” | False |

**B- Brute Force**

**-**check the to see if the first letter of the input string is a word in the dictionary, if not check the first and second letter, if not check the first, second, and third letter and so on until a word is found.

-repeat this for each word in the input string

-if every word in the input string is found in the dictionary return true

-if there is a word in the input string not found in the dictionary return false

**O- Optimize**

-Use dynamic programming to break up all combinations of the input string

**W- Walk Through**

**-**create a method that takes in the input string and a dictionary of words

**-**if the input string contains one word and that word is in the dictionary return true

-start at the first letter of the input string and check if it is in the dictionary.

-keep increasing the letters checked in the input string until a word is found in the dictionary

-if not found return false

-if found store it in a variable.

-repeat this process for the remainder of the string by calling the same method and passing in the remainder of the input string and the dictionary

**I- Implement**

public static boolean canSegment(String sentence, ArrayList<String> dictionary) {

sentence = sentence.toLowerCase();

if (dictionary.contains(sentence)) {

return true;

}

int length = sentence.length();

String finalSentence;

for (int i = 1; i<length; i++) {

String prefix = sentence.substring(0, i);

if (dictionary.contains(prefix)) {

String suffix = sentence.substring(i, length);

boolean nextWords = canSegment(suffix, dictionary);

if (nextWords != false) {

finalSentence = prefix + " " + nextWords;

return true;

}

}

}

return false;

}

**T- Test**

|  |  |  |
| --- | --- | --- |
| Class | Sample Input | Out |
| Empty dictionary | “pear” | False |
| Empty string | “” | False |
| Null | Null | Throw exception |
| Input tring that has repeated words | “girlenjoysalmonenjoysun” | True |
| Capitalized in input string but not in dictionary and vice versa | “youLeave” | True |
| Input string contains no words that are in the dictionary | “boblikescheese” | False |
| Input string contains some words that are in the dictionary | “sheleavesfootprints”  “footprintssheleaves” | False  False |
| All words in the input string are in the dictionary | “youenjoysalmon” | True |
| Input string contains non alphabetic characters | “123@salmon” | False |