

Text Justification

[My Submissions \(/problems/text-justification/submissions/\)](/problems/text-justification/submissions/)

Total

Question

Solution

Accepted:

24546Total Submissions: **165081**Difficulty: **Hard**

Given an array of words and a length L , format the text such that each line has exactly L characters and is fully (left and right) justified.

You should pack your words in a greedy approach; that is, pack as many words as you can in each line. Pad extra spaces ' ' when necessary so that each line has exactly L characters.

Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line do not divide evenly between words, the empty slots on the left will be assigned more spaces than the slots on the right.

For the last line of text, it should be left justified and no extra space is inserted between words.

For example,

words: ["This", "is", "an", "example", "of", "text", "justification."]

L: 16 .

Return the formatted lines as:

```
[
  "This    is    an",
  "example  of text",
  "justification. "
]
```

Note: Each word is guaranteed not to exceed L in length.

[click to show corner cases.](#)

[Show Tags](#)

Have you met this question in a real interview?

[Discuss \(/discuss/questions/oj/text-justification\)](/discuss/questions/oj/text-justification)

Python

[✉ Send Feedback \(mailto:admin@leetcode.com?subject=Feedback\)](mailto:admin@leetcode.com?subject=Feedback)

```
1 class Solution(object):
2     def fullJustify(self, words, maxWidth):
3         """
4         :type words: List[str]
5         :type maxWidth: int
6         :rtype: List[str]
7         """
8         returnWords=[]
9         i=0
10        while i<len(words):
11            sum=-1
12            j=i
13            while i<len(words) and sum+1+len(words[i])<=maxWidth:
14                sum+=1+len(words[i])
15                i+=1
16            tmp=words[j]
17            if i>=len(words) or i-j==1:
18                for k in range(j+1,i):
19                    tmp+=" "+words[k]
20                    tmp+=" "*(maxWidth-sum)
21            else:
22                m = (maxWidth-sum)/(i-j-1)
23                n = (maxWidth-sum)%(i-j-1)
24                for k in range(j+1,i):
25                    tmp+=" "*(m+1)
26                    if n>0:
27                        n-=1
28                        tmp+=" "
29                    tmp+=words[k]
30            returnWords.append(tmp)
31        return returnWords
```

Custom Testcase ☐

Run Code

Submit Solution

[Frequently Asked Questions \(/faq/\)](#) | [Terms of Service \(/tos/\)](#)

[Privacy](#)

Copyright © 2015 LeetCode