

Problem 68: Text Justification

Problem Information

Difficulty: Hard

Acceptance Rate: 49.69%

Paid Only: No

Tags: Array, String, Simulation

Problem Description

Given an array of strings `words` and a width `maxWidth`, format the text such that each line has exactly `maxWidth` 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 `maxWidth` characters.

Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line does 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.

Note:

* A word is defined as a character sequence consisting of non-space characters only. * Each word's length is guaranteed to be greater than `0` and not exceed `maxWidth`. * The input array `words` contains at least one word.

Example 1:

Input: `words = ["This", "is", "an", "example", "of", "text", "justification."], maxWidth = 16`

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

Example 2:

****Input:**** words = ["What", "must", "be", "acknowledgment", "shall", "be"], maxWidth = 16
****Output:**** ["What must be", "acknowledgment ", "shall be "] ****Explanation:**** Note that the last line is "shall be " instead of "shall be", because the last line must be left-justified instead of fully-justified. Note that the second line is also left-justified because it contains only one word.

****Example 3:****

****Input:**** words = ["Science", "is", "what", "we", "understand", "well", "enough", "to", "explain", "to", "a", "computer.", "Art", "is", "everything", "else", "we", "do"], maxWidth = 20 ****Output:**** ["Science is what we", "understand well", "enough to explain to", "a computer. Art is", "everything else we", "do "]

****Constraints:****

* `1 <= words.length <= 300` * `1 <= words[i].length <= 20` * `words[i]` consists of only English letters and symbols. * `1 <= maxWidth <= 100` * `words[i].length <= maxWidth`

Code Snippets

C++:

```
class Solution {
public:
    vector<string> fullJustify(vector<string>& words, int maxWidth) {

    }

};
```

Java:

```
class Solution {
    public List<String> fullJustify(String[] words, int maxWidth) {

    }

}
```

Python3:

```
class Solution:
    def fullJustify(self, words: List[str], maxWidth: int) -> List[str]:
```

