@codestorywithMIK
(Instagram, Facebook)

CSwithMIK → Twitter

codestorywithMIK

# STRINGS...

video - 18

Leetcode - 68

Hard

# Text Justification

Asked By :- Google

## 68. Text Justification

Hard 2549 3680 Add to List Share

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:**

"z"

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

spaces = 3 eachSlot = 3/2 = 1 ←

slots = 2 extra = 3%2 = 1 ←

This==is==an
example""of-text
justification.__

j - i = 3 - 0 = 3 - 1 = 2

**Example 1:**

i

j

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

This ___ ___ ___ is ___ ___ ___ ___ an ←
example ___ ___ of ___ text ___
justification. ___ ___

j →

i = 0;

while (i < n) {

lettersCount = words[i].length();

j = i + 1;

int Gaddha = 0;

while ( j < n && words[j].length() + 1
lettersCount + Gaddha
<= maxWidth) {

lettersCount += words[j].length();

→ Gaddha += 1;

j++;

}

```
int remainingspaces = maxwidth - lettersCount;

int EachGaddhaspaces = remainspaces / Gaddha;
int ExtraGaddha = remainspaces % Gaddha;

if (j == n) {//last line
    EachGaddhaspaces = 1;
    Extraspacegaps = 0;
}

string Line = findLine();

i = j;
}
```

**Example 1:**

i j

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

EachGaddhaSpace = i , ExtraspaceGaddha = i

**Example 1:**

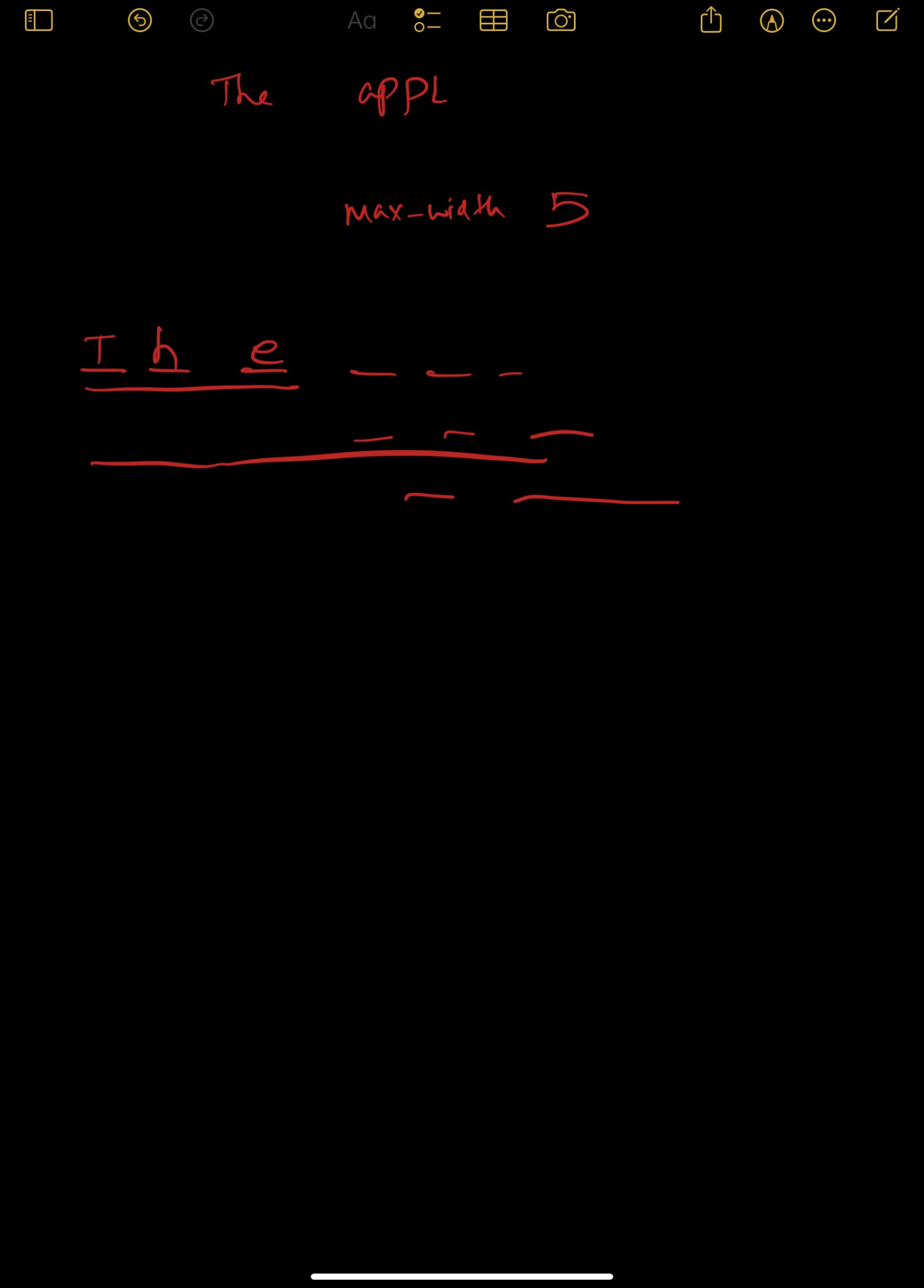
**Input:** words = ["This", "is", "an", "example", "of", "text", "justification."], maxWidth = 16

i j

EachGaddhaSpace = i , ExtraSpaceGaddha = i

Storing line ;

```
for(int K = i ; K<j ; K++) {
    line += words[K];
    for (int z = 1 ; z <= EachGaddhaSpace ; z++) {
        line += " ";
    }
    if (ExtraSpa.Ga > 0) {
        line += " ";
    }
}
while (line.length() <= maxwidth)
    line += " ";
return line ;
}
```


The appL
max-width 5
The