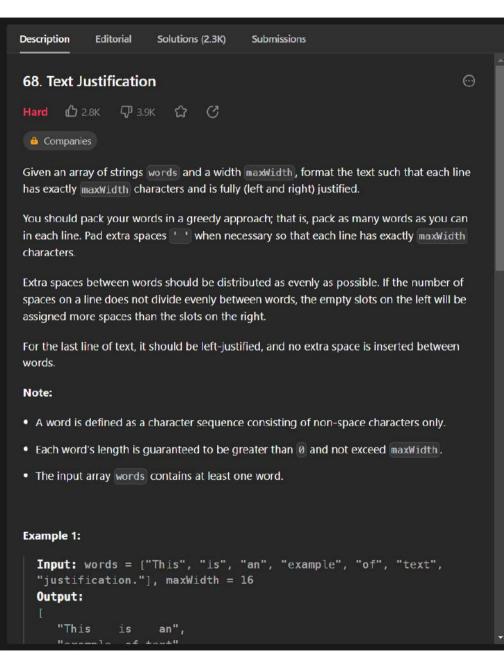


```
i Java ∨ | Auto
         public int climbStairs(int n) {
             if(n==1) return 1;
             if(n==2) return 2;
             int[] a = new int[n];
             a[0]=1;
             a[1]=2;
             for(int i=2;i<n;i++){
                 a[i]=a[i-1]+a[i-2];
             return a[n-1];
 16 }
Testcase
Accepted Runtime: 0 ms
 • Case 1
              • Case 2
Input
  2
Output
  2
Console v
                                                                            Run
                                                                                     Submit
```



```
i Java ∨ | Auto
     public class Solution {
         public List<String> fullJustify(String[] words, int maxWidth) {
             List<String> res = new ArrayList<>();
             List<String> cur = new ArrayList<>();
             int num of letters = 0;
             for (String word : words) {
                 if (word.length() + cur.size() + num of letters > maxWidth) {
                     for (int i = 0; i < maxWidth - num of letters; i++) {
                         cur.set(i % (cur.size() - 1 > 0 ? cur.size() - 1 : 1), cur.get(i % )
     (cur.size() - 1 > 0 ? cur.size() - 1 : 1)) + " ");
                     StringBuilder sb = new StringBuilder();
                     for (String s : cur) sb.append(s);
                     res.add(sb.toString());
                     cur.clear();
                     num of letters = 0;
Testcase
Accepted Runtime: 1 ms
  • Case 1
              • Case 2

    Case 3

Input
  ["This", "is", "an", "example", "of", "text", "justification."]
 maxWidth =
  16
Output
Console v
                                                                            Run
                                                                                      Submit
```