**难度: 简单**

统计字符串中的单词个数，这里的单词指的是连续的不是空格的字符。

请注意，你可以假定字符串里不包括任何不可打印的字符。

示例:

输入: "Hello, my name is John"

输出: 5

解释: 这里的单词是指连续的不是空格的字符，所以 "Hello," 算作 1 个单词。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/number-of-segments-in-a-string

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

思路1(利用正则):

1. 先去掉前后的空格调用字符串自带的trim方法

2. 判断一下去掉前后空格的长度是否是0,是直接返回0

3. 使用正则通过一个空格或多个空格来进行切割然后获取长度,返回

|  |
| --- |
| public int countSegments(String s) {  //去掉前后空格  String trim = s.trim();  if (trim.length() == 0) {  return 0;  }  //直接使用正则按空格一个或多个进行切割,然后获取它的长度  return trim.split("\\s+").length;  } |

思路2:

1. 先去掉前后的空格调用字符串自带的trim方法

2. 判断一下去掉前后空格的长度是否是0,是直接返回0

3.反着判断

|  |
| --- |
| public int countSegments2(String s) {  //去掉前后空格  String trim = s.trim();  if (trim.length() == 0) {  return 0;  }  //计数  int count = 0;  for(int i = 0; i < trim.length(); i++) {  //反着判断...当我们i-1是空格,i不是空格就说明是单词  if ( (i == 0 || trim.charAt(i-1) == ' ') && trim.charAt(i) != ' ') {  count++;  }  }  return count;  } |