Given a string *s* consists of upper/lower-case alphabets and empty space characters ' ', return the length of last word in the string.

If the last word does not exist, return 0.

**Note:** A word is defined as a character sequence consists of non-space characters only.

**Example:**

**Input:** "Hello World"

**Output:** 5

问题：返回最后一个单词的长度，每个单词用空格隔开。

要是不用java本身的类库来说，思路是我们把s转换成字符数组，从后边开始遍历，遍历到最近的一个空格停止。

代码

public static int lengthOfLastWord(String s) {

int count = 0;

for (int i = s.length() - 1; i >= 0 && s.charAt(i) != ' '; i--) {

count++;

}

return count;

}

我自己的想法是，用String.split()方法把字符串分割开，返回子字符串数组，然后去最后一个，遍历最后一个子字符串进行计数。

代码

public static int lengthOfLastWord(String s){

String strs[] = s.split(" ");

int count = 0;

for(char c : strs[strs.length-1].toCharArray()){

count++;

}

return count;

}

看了答案，答案有个很nb的，就是用s的长度减去最后一个空格的长度-1，就是最后一个子字符串的长度，这里用的到一个lastIndexOf()方法就是去某个位于最后的子字符串的位置。顺便提String.trim()的用途就是将字符串多余出的位置取消掉，把当前String长度更正为正好和内容相等的长度。

public static int lengthOfLastWord(String s) {

return (s.trim().length()-1)-**s.trim().lastIndexOf(" ")**;

}