Given a string `s` consisting of words and spaces, return *the length of the **last** word in the string.*

A **word** is a maximal substring consisting of non-space characters only.

Example 1:

```
Input: s = "Hello World"
Output: 5
Explanation: The last word is "World" with length 5.
```

Example 2:

```
Input: s = "   fly me   to   the moon  "
Output: 4
Explanation: The last word is "moon" with length 4.
```

Example 3:

```
Input: s = "luffy is still joyboy"
Output: 6
Explanation: The last word is "joyboy" with length 6.
```

Constraints:

- `1 <= s.length <= 10^4`
- `s` consists of only English letters and spaces `' '`.
- There will be at least one word in `s`.

Python:

```
class Solution:
    def lengthOfLastWord(self, s: str) -> int:
        length = 0
        i = len(s) - 1

        # Skip trailing spaces
        while i >= 0 and s[i] == ' ':
            i -= 1

        # Count characters of last word
        while i >= 0 and s[i] != ' ':
```

```
            length += 1
            i -= 1

        return length
        # length = 0
        # i = len(s) - 1

        # # Skip trailing spaces
        # while i >= 0 and s[i] == ' ':
        #     i -= 1

        # # Count characters of last word
        # while i >= 0 and s[i] != ' ':
        #     length += 1
        #     i -= 1

        # return length
```

JavaScript:

```
/**
 * @param {string} s
 * @return {number}
 */
var lengthOfLastWord = function(s) {
    // Remove leading/trailing spaces
    let words = s.trim().split(" ");

    // Get the last word
    let lastWord = words[words.length - 1];

    // Return its length
    return lastWord.length;
};
```

Java:

```
class Solution {
    public int lengthOfLastWord(String s) {
        int length = 0;
        int i = s.length() - 1;

        // Skip trailing spaces
        while (i >= 0 && s.charAt(i) == ' ') {
            i--;
        }
```

```
        // Count the last word length
        while (i >= 0 && s.charAt(i) != ' ') {
            length++;
            i--;
        }

        return length;
    }
}
```