🔷 Problem Statement (Simple Words)

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

👉 A **word** = continuous characters without spaces.

Example

Input: "Hello World"

Output: 5

Last word = "World" → length = **5**

✅ APPROACH 1: Using split() method

💡 Idea

1. Remove extra spaces using trim()
2. Split string into words using space " "
3. Return length of last word

□ Example

s = "   Fly me   to   the moon "

Step-by-step

1. trim()

"Fly me   to   the moon"

2. split(" ")

["Fly", "me", "to", "the", "moon"]

3. Last word = "moon"
4. Length = **4**

□ Dry Run

Step	Operation	Result
1	trim()	"Fly me to the moon"
2	split()	["Fly","me","to","the","moon"]
3	last word	"moon"

Step	Operation	Result
4	length	4

⏱️ Complexity

- **Time:** O(n)
- **Space:** O(n) (extra array)

💻 Code – Split Method

Java

```
class Solution {
    public int lengthOfLastWord(String s) {
        s = s.trim();
        String[] words = s.split(" ");
        return words[words.length - 1].length();
    }
}
```

C++

```
class Solution {
public:
    int lengthOfLastWord(string s) {
        stringstream ss(s);
        string word, last;
        while (ss >> word) {
            last = word;
        }
        return last.length();
    }
};
```

Python

```
class Solution:
```

```
def lengthOfLastWord(self, s: str) -> int:
    words = s.strip().split()
    return len(words[-1])
```

✅ **APPROACH 2: Traverse from Back (Optimized)**

💡 **Idea**

1. Start from **end of string**
2. Skip trailing spaces
3. Count characters until space appears

🚀 **No extra space used**

▯ **Example**

s = "Hello World   "

▯ **Dry Run (Character by Character)**

Index	Char	Action	Count
11	' '	skip	0
10	' '	skip	0
9	'd'	count	1
8	'l'	count	2
7	'r'	count	3
6	'o'	count	4
5	'W'	count	5
4	' '	stop	return 5

⏱ **Complexity**

- **Time:** O(n)
- **Space:** O(1) ✅

💻 **Code – Backward Traversal**

Java

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

        while (i >= 0 && s.charAt(i) == ' ') {
            i--;
        }

        while (i >= 0 && s.charAt(i) != ' ') {
            count++;
            i--;
        }

        return count;
    }
}
```

C++

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

        while (i >= 0 && s[i] == ' ') i--;

        while (i >= 0 && s[i] != ' ') {
            count++;
            i--;
        }
        return count;
    }
};
```

Python

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

        while i >= 0 and s[i] == ' ':
            i -= 1

        while i >= 0 and s[i] != ' ':
            count += 1
            i -= 1

        return count
```

🔥 Final Comparison

Approach	**Time**	**Space**	**Best Use**
Split Method	O(n)	O(n)	Easy to understand
Backward Traverse	O(n)	O(1)	**Interview preferred**