
Length of Last Word

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.

For example,

Given $s = \text{"Hello World"}$,
return 5.

Solution 1

I've noticed that a lot of solutions use available library functions that return directly the positions of certain characters or do other operations like "split". I personally don't think that's a good idea. Firstly, these functions take some time and usually involve with iteration through the whole string. Secondly, questions like this one is intended to be a practice of detail implementation, not calling other functions. My solution like below uses only the most basic string operations and probably beats many other solutions which call other existing functions.

```
int lengthOfLastWord(const char* s) {  
    int len = 0;  
    while (*s) {  
        if (*s++ != ' '){  
            ++len;  
        } else if (*s && *s != ' '){  
            len = 0;  
        }  
    }  
    return len;  
}
```

written by [eaglesky1990](#) original link [here](#)

Solution 2

```
public int lengthOfLastWord(String s) {  
    return s.trim().length()-s.trim().lastIndexOf(" ")-1;  
}
```

written by [lvlolitte](#) original link [here](#)

Solution 3

Well, the basic idea is very simple. Start from the tail of **s** and move backwards to find the first non-space character. Then from this character, move backwards and count the number of non-space characters until we pass over the head of **s** or meet a space character. The count will then be the length of the last word.

```
class Solution {
public:
    int lengthOfLastWord(string s) {
        int len = 0, tail = s.length() - 1;
        while (tail >= 0 && s[tail] == ' ') tail--;
        while (tail >= 0 && s[tail] != ' ') {
            len++;
            tail--;
        }
        return len;
    }
};
```

written by [jianchao.li.fighter](#) original link [here](#)

From [LeetCoder](#).