Previous Next

434. Number of Segments in a String

Nov. 3, 2017 | 18.1K views

Average Rating: 4.20 (10 votes)

Count the number of segments in a string, where a segment is defined to be a contiguous sequence of non-space characters.

Please note that the string does not contain any **non-printable** characters.

Example:

```
Input: "Hello, my name is John"
Output: 5
```

Approach #1 Using Language Builtins [Accepted]

Intuition

In a situation where raw efficiency is less important than code legibility, it is likely better to use language-idiomatic builtin functions to solve this problem.

Algorithm

There are a few corner cases that you can get snagged on in this problem, at least in Java. First, one or more leading spaces will cause `split` to deduce an erroneous `""` token at the beginning of the string, so we use the builtin `trim` method to remove leading and trailing spaces. Then, if the resulting string is the empty string, then we can simply output `0`. This is necessary due to the following behavior of the `split` method:

```
String[] tokens = "".split("\\s++");
tokens.length; // 1
tokens[0]; // ""
```

If we reach the final return statement, we `split` the trimmed string on sequences of one or more whitespace characters (`split` can take a regular expression) and return the length of the resulting array.

The Python solution is trivially short because Python's `split` has a lot of default behavior that makes it perfect for this sort of problem. Notably, it returns an empty list when `split`ting an empty string, it splits on whitespace by default, and it implicitly `trim`s (`strip`s, in Python lingo) the string beforehand.

Java | Python | Copy

```python
class Solution:
    def countSegments(self, s):
        return len(s.split())
```

Complexity Analysis

- Time complexity : $\mathcal{O}(n)$

  All builtin language functionality used here (in both the Java and Python examples) runs in either $\mathcal{O}(n)$ or $\mathcal{O}(1)$ time, so the entire algorithm runs in linear time.

- Space complexity : $\mathcal{O}(n)$

  `split` (in both languages) returns an array/list of $\mathcal{O}(n)$ length, so the algorithm uses linear additional space.

Approach #2 In-place [Accepted]

Intuition

If we cannot afford to allocate linear additional space, a fairly simple algorithm can deduce the number of segments in linear time and constant space.

Algorithm

To count the number of segments, it is equivalent to count the number of string indices at which a segment begins. Therefore, by formally defining the characteristics of such an index, we can simply iterate over the string and test each index in turn. Such a definition is as follows: a string index begins a segment if it is preceded by whitespace (or is the first index) and is not whitespace itself, which can be checked in constant time. Finally, we simply return the number of indices for which the condition is satisfied.

Java | Python | Copy

```python
class Solution:
    def countSegments(self, s):
        segment_count = 0

        for i in range(len(s)):
            if (i == 0 or s[i-1] == ' ') and s[i] != ' ':
                segment_count += 1

        return segment_count
```

Complexity Analysis

- Time complexity : $\mathcal{O}(n)$

  We do a constant time check for each of the string's n indices, so the runtime is overall linear.

- Space complexity : $\mathcal{O}(1)$

  There are only a few integers allocated, so the memory footprint is constant.

Rate this article: ★★★★★

Previous Next

Comments: 14 Sort By

Type comment here... (Markdown is supported)

Preview Post

frankgt401 ★ 26 August 16, 2018 11:39 AM
What if the characters contains '\t' or '\n'?
9 Share Reply
SHOW 2 REPLIES

panyifei ★ 15 May 30, 2018 9:37 AM
Why we cannot using trimmed.split(" ") directly, thanks!!!!!
8 Share Reply
SHOW 2 REPLIES

dev_ps ★ 14 January 20, 2019 5:26 PM
c++ solution, 0 ms

```
int countSegments(string s) {

int c=0,n=s.size();
```

Read More
5 Share Reply

ACQuester ★ 13 May 3, 2019 10:42 AM
C++ Language Builtins

```
class Solution {
public:
    int countSegments(string s) {
```

Read More
2 Share Reply

BSpwr ★ 0 January 16, 2019 12:10 AM
C++ solution, 0ms

```
class Solution {
public:
    int countSegments(string s) {
```

Read More
0 Share Reply

hongkong20170701 ★ 464 November 20, 2018 3:11 PM
[Java] Reading it clearly is more difficult than coding. bad!

```
class Solution {
    public int countSegments(String s) {
        if (s == null || s.trim().equals("")) return 0;
```

Read More
0 Share Reply

prakashmanwani ★ 2 February 27, 2018 1:33 AM
Simple easy to understand java solution

```
class Solution {
    public int countSegments(String s) {
        int counter = 0;
```

Read More
0 Share Reply

nwadhwa12345 ★ 34 January 23, 2018 6:53 AM
My easy solution in java

class Solution {
public int countSegments(String s) {
int count =0;

Read More
0 Share Reply

antzshrek ★ 0 December 15, 2017 7:54 AM
well written!
0 Share Reply

leowang7 ★ 3 July 7, 2020 8:11 PM
python solutions:

```
def countSegments(self, s: str) -> int:
    s_new = s.strip().split(' ')
    cnt = 0
```

Read More
0 Share Reply

‹ 1 2 ›