

目录

第 1 章	String	1	1.2	Reverse String II	2
1.1	Reverse String	1	1.3	Reverse Words in a String	3

第 1 章

String

1.1 Reverse String

Description

Write a function that takes a string as input and returns the string reversed.

Example:

Given s = "hello", return "olleh".

Solution

```
public String reverseString(String s) {  
    return new StringBuilder(s).reverse().toString();  
}
```

1.2 Reverse String II

Description

Given a string and an integer k , you need to reverse the first k characters for every $2k$ characters counting from the start of the string. If there are less than k characters left, reverse all of them. If there are less than $2k$ but greater than or equal to k characters, then reverse the first k characters and left the other as original.

Example:

Input: $s = \text{"abcdefg"}, k = 2$

Output: "bacdfeg"

Restrictions:

The string consists of lower English letters only.

Length of the given string and k will in the range $[1, 10000]$

Solution

```
public String reverseStr(String s, int k) {  
    if (s.length() <= k) {  
        return new StringBuilder(s).reverse().toString();  
    }  
    if (s.length() <= 2 * k) {  
        return reverseStr(s.substring(0, k), k) + s.substring(k);  
    }  
    return reverseStr(s.substring(0, 2 * k), k) + reverseStr(s.substring(2 * k), k);  
}
```

1.3 Reverse Words in a String

Description

Given an input string, reverse the string word by word.

For example, Given s = "the sky is blue", return "blue is sky the".

Clarification:

1. What constitutes a word?

A sequence of non-space characters constitutes a word.

2. Could the input string contain leading or trailing spaces?

Yes. However, your reversed string should not contain leading or trailing spaces.

3. How about multiple spaces between two words?

Reduce them to a single space in the reversed string.

Solution I

```
// 耗时 10ms
public String reverseWords(String s) {
    StringBuilder sb = new StringBuilder();

    boolean inWord = false;
    char[] cc = s.toCharArray();
    for (int i = cc.length - 1, idx = 0; i >= 0; i--) {
        if (cc[i] != ' ') {
            if (!inWord && sb.length() > 0) {
                sb.append(' ');
            }
            inWord = true;
            sb.insert(idx, cc[i]);
        } else if (inWord) {
            idx = sb.length() + 1;
            inWord = false;
        }
    }

    return sb.toString();
}
```

Solution II

```
// 耗时 18ms
public String reverseWords2(String s) {
    StringBuilder sb = new StringBuilder();

    for (int i = 0, j = 0; i < s.length(); ) {
        if (s.charAt(i) == ' ') {
            i++;
            j = i;
        } else if (j >= s.length() || s.charAt(j) == ' ') {
            sb.insert(0, s.substring(i, j) + " ");
            i = j;
        } else {
            j++;
        }
    }
}
```

```
        }  
    }  
  
    if (sb.length() > 0) {  
        sb.setLength(sb.length() - 1);  
    }  
  
    return sb.toString();  
}
```