# 【LeetCode】344. Reverse String 解题报告

# 目录[(?)[-]](http://blog.csdn.net/crazy1235/article/details/51420522)

1. [Subject](http://blog.csdn.net/crazy1235/article/details/51420522#t0)
2. [Explain](http://blog.csdn.net/crazy1235/article/details/51420522#t1)
3. [Solution](http://blog.csdn.net/crazy1235/article/details/51420522#t2)
   1. [solution 1](http://blog.csdn.net/crazy1235/article/details/51420522#t3)
   2. [solution 2](http://blog.csdn.net/crazy1235/article/details/51420522#t4)
   3. [solution 3](http://blog.csdn.net/crazy1235/article/details/51420522#t5)
   4. [solution 4](http://blog.csdn.net/crazy1235/article/details/51420522#t6)
   5. [solution 5](http://blog.csdn.net/crazy1235/article/details/51420522#t7)
   6. [solution 6](http://blog.csdn.net/crazy1235/article/details/51420522#t8)

## Subject

出处：<https://leetcode.com/problems/reverse-string/>

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

Example:   
Given s = “hello”, return “olleh”.

## Explain

该题目的意思就是一个简单的**字符串反转**输出。

so easy~~

## Solution

### solution 1

将string转换成**char[] 数组**。然后for循环从数组末尾开始向前整合。

public static String reverseString(String s) {

String result = "";

char[] ch = s.toCharArray();

for (int i = ch.length - 1; i >= 0; i--) {

result += ch[i];

}

return result;

}

public static String reverseString2(String s) {

String result = "";

for (int i = s.length() - 1; i >= 0; i--) {

result += s.charAt(i);

}

return result;

}

### solution 2

直接使用**StringBuffer**类的**reverse()**方法。

public static String reverseString3(String s) {

return new StringBuffer(s).reverse().toString();

}

### solution 3

solution 1的时间复杂度都是o(n).

实际上我们可以for循环”一半”数据长度。将**首位字符交换**即可。

此时时间复杂度为**o(n/2)**.

public static String reverseString4(String s) {

char[] ch = s.toCharArray();

int halfLength = s.length() / 2;

char temp;

for (int i = 0; i < halfLength; i++) {

temp = ch[s.length() - 1 - i];

ch[s.length() - 1 - i] = ch[i];

ch[i] = temp;

}

return new String(ch);

}

### solution 4

第四种方案采用【**异或运算**】。

**异或预算满足交换律**

关于异或运算的性质及应用，可以参考这篇blog：

<http://www.cnblogs.com/suoloveyou/archive/2012/04/25/2470292.html>

public static String reverseString5(String s) {

char[] ch = s.toCharArray();

int start = 0;

int end = ch.length - 1;

while (start < end) {

ch[start] = (char) (ch[start] ^ ch[end]);

ch[end] = (char) (ch[start] ^ ch[end]);

ch[start] = (char) (ch[start] ^ ch[end]);

start++;

end--;

}

return new String(ch);

}

该方案的时间复杂度也是**o(n/2)**.

### solution 5

通过**栈**来做。不过有点大材小用的感觉。O(∩\_∩)O~

public static String reverseString6(String s) {

Stack<Character> stack = new Stack<>();

char[] ch = s.toCharArray();

String result = "";

for (int i = 0; i < ch.length; i++) {

stack.push(ch[i]);

}

for (int i = 0; i < ch.length; i++) {

result += stack.pop();

}

return result;

}

时间复杂度是**o(2n)**.

### solution 6

通过递归的方式来做。

public static String reverseString7(String s) {

int length = s.length();

if (length <= 1) {

return s;

}

String leftStr = s.substring(0, length / 2);

String rightStr = s.substring(length / 2, length);

return reverseString7(rightStr) + reverseString7(leftStr);

}