**难度: 简单**

编写一个函数，其作用是将输入的字符串反转过来。输入字符串以字符数组 char[] 的形式给出。

不要给另外的数组分配额外的空间，你必须原地修改输入数组、使用 O(1) 的额外空间解决这一问题。

你可以假设数组中的所有字符都是 ASCII 码表中的可打印字符。

示例 1：

输入：["h","e","l","l","o"]

输出：["o","l","l","e","h"]

示例 2：

输入：["H","a","n","n","a","h"]

输出：["h","a","n","n","a","H"]

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/reverse-string

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

思路1(双支针while循环):

1). 第一个和最后互换,第二个与最后一个的前一个互换

2). 依次类推,知道左边的等于右边

**互换不使用第三个变量,省空间**

|  |
| --- |
| public void reverseString(char[] s) {  int l = 0;  int r = s.length - 1;  while(l < r) {  s[l] = (char)(s[l] + s[r]);  s[r] = (char)(s[l] - s[r]);  s[l] = (char)(s[l] - s[r]);  l++;  r--;  }  } |

思路2(双支针for循环):

1). 第一个和最后互换,第二个与最后一个的前一个互换

2). 依次类推,知道左边的等于右边

**互换不使用第三个变量,省空间**

|  |
| --- |
| public void reverseString2(char[] s) {  for (int l = 0, r = s.length -1; l < r; l++,r--) {  s[l] = (char)(s[l] + s[r]);  s[r] = (char)(s[l] - s[r]);  s[l] = (char)(s[l] - s[r]);  }  } |

**思路1: 使用多一个辅助变量的代码(思路2也可以同样增多一个辅助变量)**

|  |
| --- |
| public void reverseString(char[] s) {  int l = 0;  int r = s.length - 1;  while(l < r) {  char temp = s[l];  s[l] = s[r];  s[r] = temp;  l++;  r--;  }  } |