

字符串 (String)

面试题 01.09. 字符串轮转

字符串轮转。给定两个字符串 `s1` 和 `s2`，请编写代码检查 `s2` 是否为 `s1` 旋转而成（比如，`waterbottle` 是 `erbottlewat` 旋转后的字符串）。

示例1:

```
输入: s1 = "waterbottle", s2 = "erbottlewat"  
输出: True
```

示例2:

```
输入: s1 = "aa", "aba"  
输出: False
```

提示:

1. 字符串长度在[0, 100000]范围内。

说明:

1. 你能只调用一次检查子串的方法吗?

■ 在有些面试题中，也称`s1`、`s2`互为旋转词

S1			
w	a	k	e

S2			
a	k	e	w

S3			
k	e	w	a

S4			
e	w	a	k

S1 + S1							
w	a	k	e	w	a	k	e

572. 另一个树的子树

给定两个非空二叉树 **s** 和 **t**，检验 **s** 中是否包含和 **t** 具有相同结构和节点值的子树。**s** 的一个子树包括 **s** 的一个节点和这个节点的所有子孙。**s** 也可以看做它自身的一棵子树。

示例 1:
给定的树 **s**:



给定的树 **t**:



返回 **true**，因为 **t** 与 **s** 的一个子树拥有相同的结构和节点值。

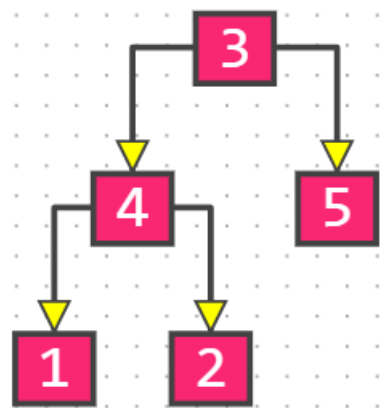
示例 2:
给定的树 **s**:



给定的树 **t**:

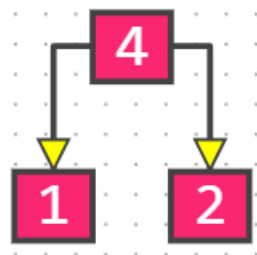


返回 **false**。



序列化 (后序遍历)

#!#!1!#!#!2!4!#!#!5!3!

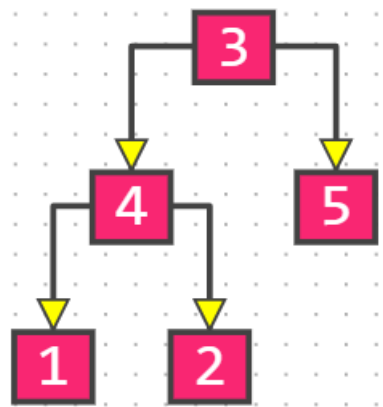


序列化 (后序遍历)

#!#!1!#!#!2!4!

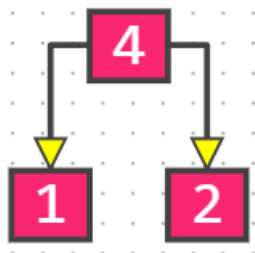
- 非空节点: 值!, 空节点: #!
- 空节点也必须要序列化, 才能完整地表达唯一的一棵树

思考：如何反序列化？



反序列化（后序遍历）

#!#!1!#!#!2!4!#!#!5!3!



反序列化（后序遍历）

#!#!1!#!#!2!4!

242. 有效的字母异位词

给定两个字符串 s 和 t ，编写一个函数来判断 t 是否是 s 的字母异位词。

示例 1:

```
输入: s = "anagram", t = "nagaram"  
输出: true
```

示例 2:

```
输入: s = "rat", t = "car"  
输出: false
```

说明:

你可以假设字符串只包含小写字母。

进阶:

如果输入字符串包含 unicode 字符怎么办？你能否调整你的解法来应对这种情况？

■ 相似的题目

□ [49. 字母异位词分组](#)

□ [面试题 10.02. 变位词组](#)

□ [438. 找到字符串中所有字母异位词](#)

■ 思路

□ 分别统计2个单词中每个字符的数量

□ 哈希表（空间换时间）

151. 翻转字符串里的单词

给定一个字符串，逐个翻转字符串中的每个单词。

输入: "the sky is blue"

输出: "blue is sky the"

输入: " hello world! "

输出: "world! hello"

解释: 输入字符串可以在前面或者后面包含多余的空格，但是反转后的字符不能包括。

输入: "a good example"

输出: "example good a"

解释: 如果两个单词间有多余的空格，将反转后单词间的空格减少到只含一个。

说明:

- 无空格字符构成一个单词。
- 输入字符串可以在前面或者后面包含多余的空格，但是反转后的字符不能包括。
- 如果两个单词间有多余的空格，将反转后单词间的空格减少到只含一个。

进阶:

请选用 C 语言的用户尝试使用 $O(1)$ 额外空间复杂度的原地解法。

■ 相似的题目: [面试题58 - I. 翻转单词顺序](#)

消除字符串中的多余空格

i
↓

0	1	2	3	4	5	6	7	8	9	10	11	12	13
		a	r	e			y	o	u		o	k	

↑
cur

space =

0	1	2	3	4	5	6	7	8	9	10	11	12	13
a	r	e		y	o	u		o	k				

0	1	2	3	4	5	6	7	8	9	10	11	12	13
a	r	e		y	o	u		o	k				

$[0, 10)$ 逆序

0	1	2	3	4	5	6	7	8	9	10	11	12	13
k	o		u	o	y		e	r	a				

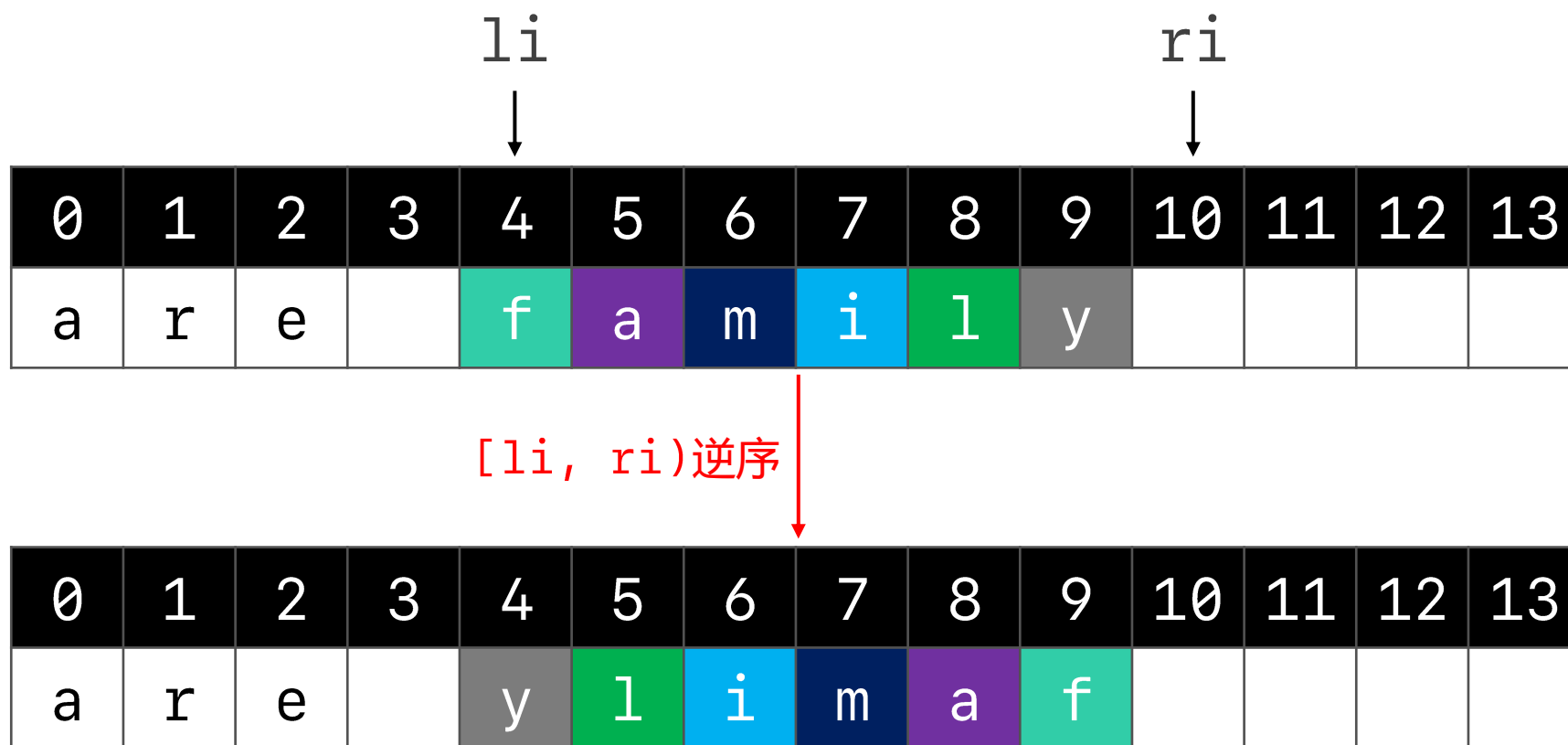
$[0, 2)$ 逆序

$[3, 6)$ 逆序

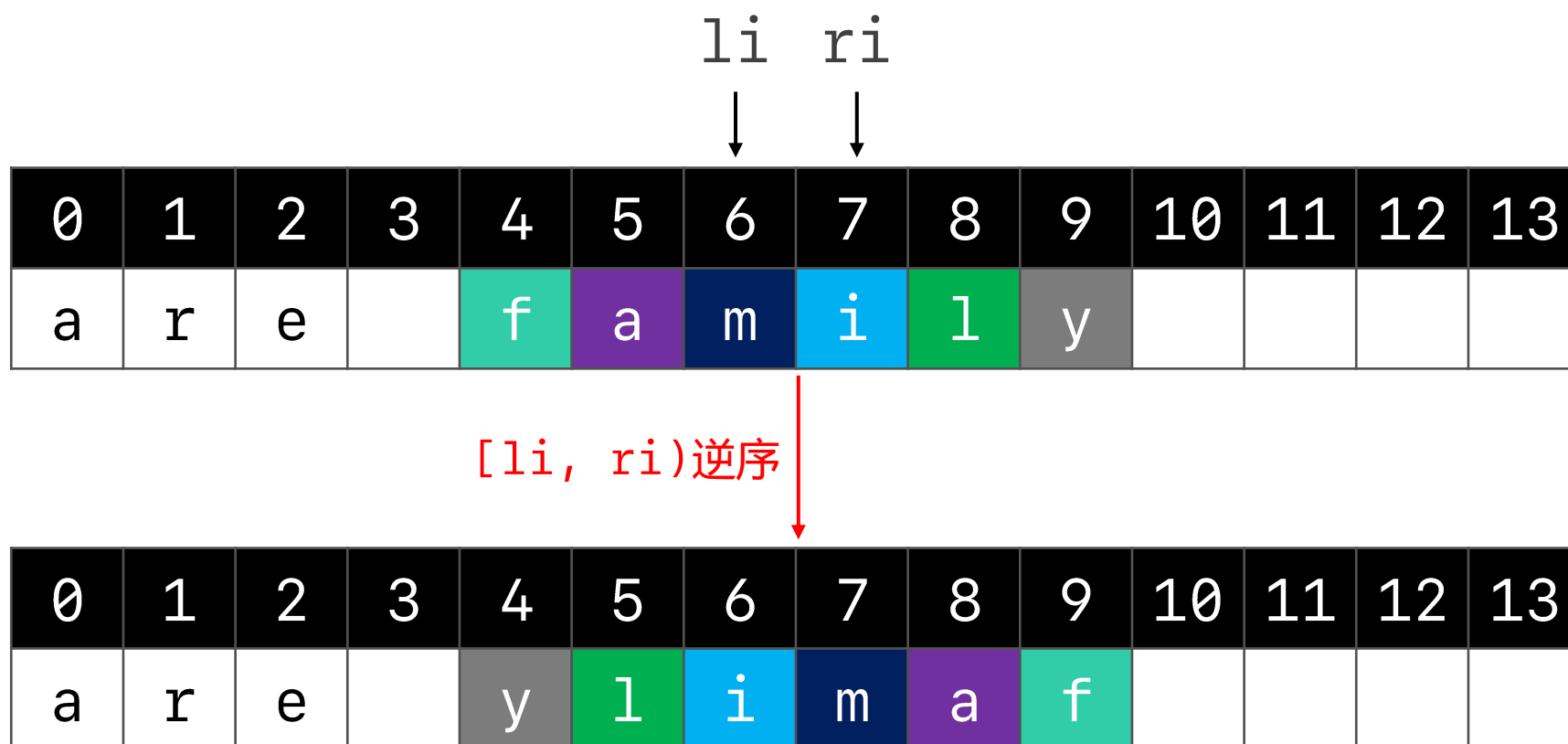
$[7, 10)$ 逆序

0	1	2	3	4	5	6	7	8	9	10	11	12	13
o	k		y	o	u		a	r	e				

指定范围的字符串逆序



指定范围的字符串逆序



3. 无重复字符的最长子串

给定一个字符串，请你找出其中不含有重复字符的 **最长子串** 的长度。

输入: "abcabcbb"

输出: 3

解释: 因为无重复字符的最长子串是 "abc", 所以其长度为 3。

输入: "bbbbbb"

输出: 1

解释: 因为无重复字符的最长子串是 "b", 所以其长度为 1。

输入: "pwwkew"

输出: 3

解释: 因为无重复字符的最长子串是 "wke", 所以其长度为 3。

请注意, 你的答案必须是 **子串** 的长度, "pwke" 是一个 **子序列**, 不是子串。

0	1	2	3	4	5
p	w	w	k	e	w

位置	字符	以这个字符结尾的最长无重复子串	以这个字符结尾的最长无重复子串的长度
0	p	p	1
1	w	pw	2
2	w	w	1
3	k	wk	2
4	e	wke	3
5	w	kew	3

pi是s[i]字符上一次出现的位置

li是以s[i-1]字符结尾的最长不重复子串的开始索引（最左索引）

0				pi		li		i-1	i	
				D				A	D	

0		li		pi				i-1	i	
				D				A	D	

0				li pi				i-1	i	
				D				A	D	

思考题

■ [5. 最长回文子串](#)

■ [72. 编辑距离](#)

作业

■ [1143. 最长公共子序列](#) ([第二季](#)中讲过)

■ [32. 最长有效括号](#)

■ [1048. 最长字符串链](#)