

算法与数据结构体系课程

liuyubobobo

更多关于归并排序法

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

归并排序算法的优化

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

必要时才 merge

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

实践：必要时才 merge

归并排序法的复杂度分析

如果 merge 不执行

arr[0...7] $T=8$

递归树

arr[0...3] $T=4$

arr[4...7] $T=4$

arr[0...1] $T=2$

arr[2...3] $T=2$

arr[4...5] $T=2$

arr[6...7] $T=2$

arr[0] $T=1$ arr[1] $T=1$

arr[2] $T=1$ arr[3] $T=1$

arr[4] $T=1$ arr[5] $T=1$

arr[6] $T=1$ arr[7] $T=1$

归并排序法的复杂度分析

如果 merge 不执行

有多少个节点?

arr[0...7] T=1

递归树

arr[0...3] T=1

arr[4...7] T=1

arr[0...1] T=1

arr[2...3] T=1

arr[4...5] T=1

arr[6...7] T=1

arr[0] T=1 arr[1] T=1

arr[2] T=1 arr[3] T=1

arr[4] T=1 arr[5] T=1

arr[6] T=1 arr[7] T=1

归并排序法的复杂度分析

非叶子节点: $n/2 + n/4 + n/8 + \dots + 1$

$$= \frac{\frac{n}{2}(1 - (\frac{1}{2})^m)}{\frac{1}{2}} < n = O(n)$$

叶子节点: n

归并排序法的复杂度分析

如果 merge 不执行

有多少个节点?

$O(n)$

arr[0...7] $T=1$

递归树

arr[0...3] $T=1$

arr[4...7] $T=1$

arr[0...1] $T=1$

arr[2...3] $T=1$

arr[4...5] $T=1$

arr[6...7] $T=1$

arr[0] $T=1$

arr[1] $T=1$

arr[2] $T=1$

arr[3] $T=1$

arr[4] $T=1$

arr[5] $T=1$

arr[6] $T=1$

arr[7] $T=1$

归并排序法的复杂度分析

对 merge 进行判断后，对完全有序的数组

归并排序是 $O(n)$ 的

插入排序法对完全有序的数组是 $O(n)$ 的

使用插入排序法优化

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

实践：使用插入排序优化

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

归并排序算法的内存优化

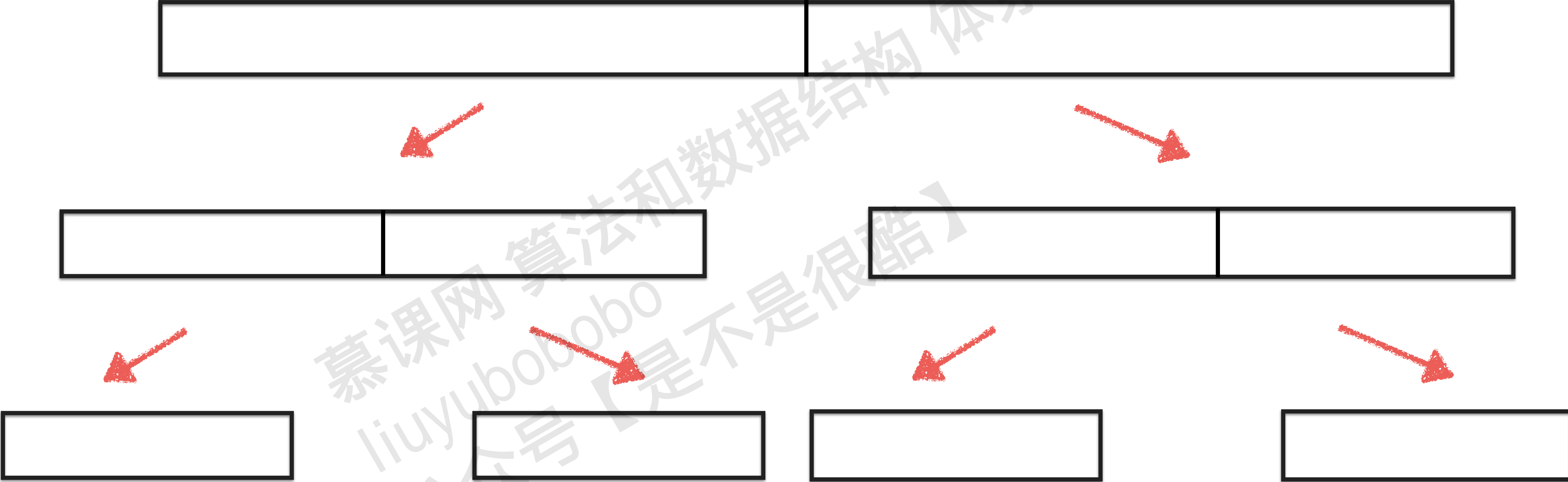
慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

实践：使用统一的 temp

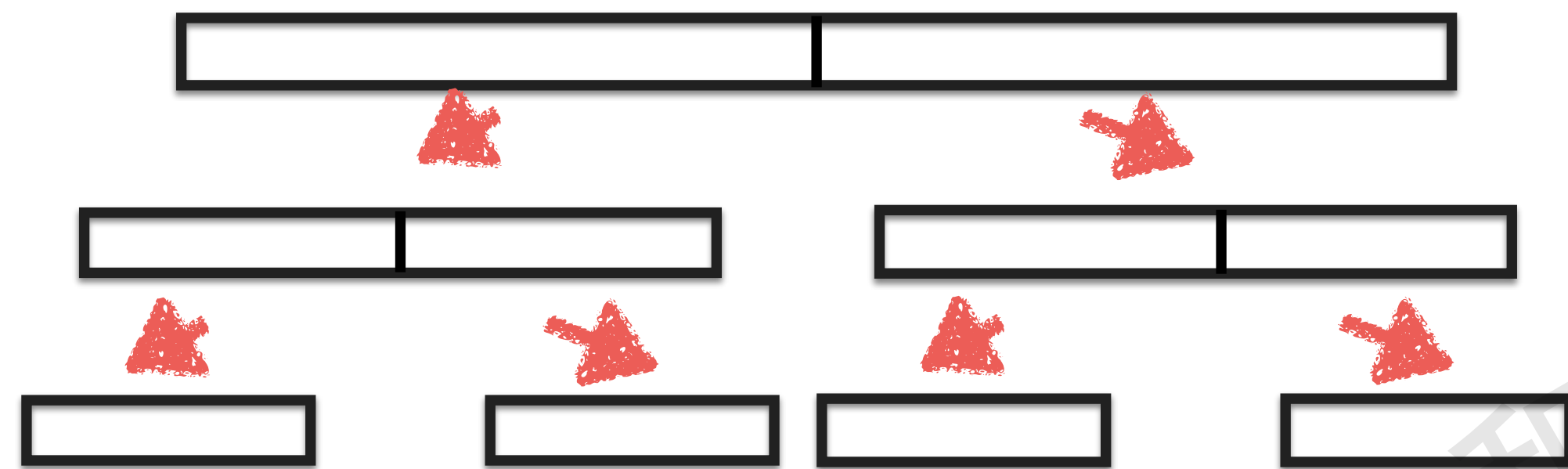
自底向上的归并排序算法

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

归并排序法



归并排序法



注意递归函数的“宏观”语意

`MergeSort(arr, l, r)`

对 `arr` 的 `[l, r]` 部分排序

```
MergeSort(arr, l, r){
```

```
    if(l >= r) return;
```

```
    int mid = (l + r) / 2;
```

```
    // 对 arr[l, mid] 进行排序
```

```
    MergeSort(arr, l, mid);
```

```
    // 对 arr[mid + 1, r] 进行排序
```

```
    MergeSort(arr, mid + 1, r);
```

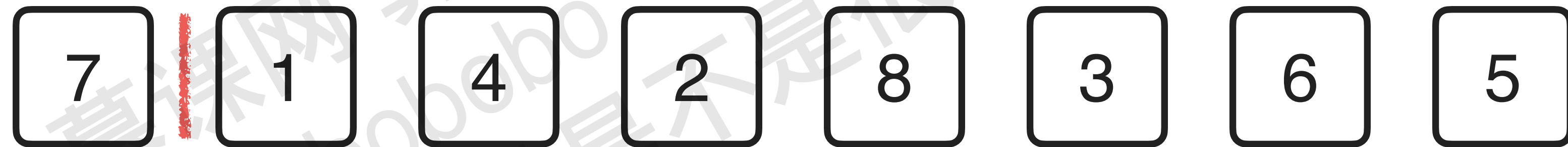
```
    // 将arr[l,mid]和arr[mid+1,r]合并
```

```
    merge(arr, l, mid, r);
```

```
}
```

自底向上归并排序

$sz = 1$



自底向上归并排序

$sz = 1$



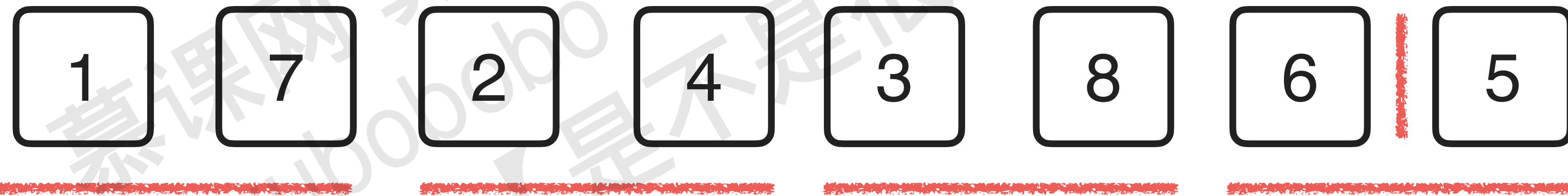
自底向上归并排序

$sz = 1$



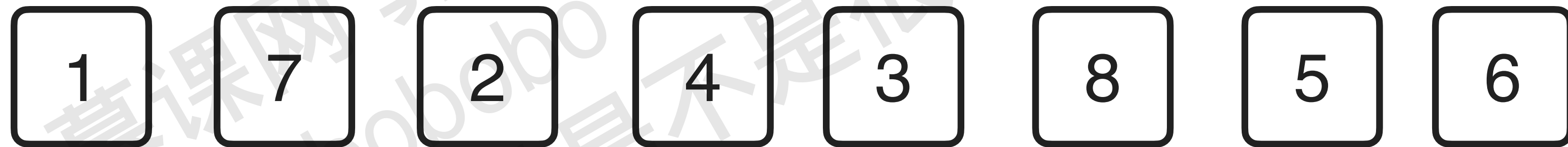
自底向上归并排序

$sz = 1$



自底向上归并排序

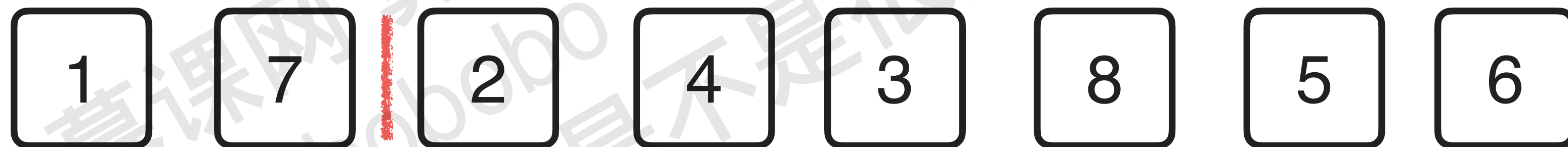
$sz = 1$



自底向上归并排序

$sz = 1$

$sz = 2$



自底向上归并排序

$sz = 1$

$sz = 2$



自底向上归并排序

$sz = 1$

$sz = 2$

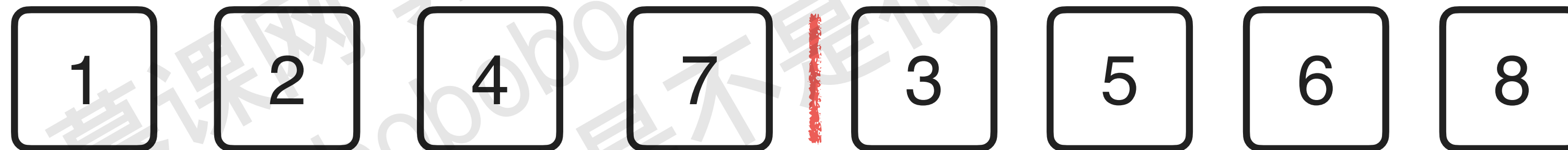


自底向上归并排序

SZ = 1

SZ = 2

SZ = 4



自底向上归并排序

$SZ = 1$

$SZ = 2$

$SZ = 4$



自底向上归并排序

SZ = 1

SZ = 2

SZ = 4

SZ = 8



实现自底向上的归并排序

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

实践：实现自底向上的归并排序

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

作业：使用插入排序
优化自底向上的归并排序

慕课网 算法和数据结构体系课程
liuyubobobo
公众号【是不是很有趣】

作业解析：使用插入排序 优化自底向上的归并排序

慕课网 算法和数据结构体系课程
liuyubobobo
公众号【是不是很有趣】

逆序数对个数问题

慕课网 算法与数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

逆序数对个数



逆序数对个数



逆序数对个数



逆序数对个数

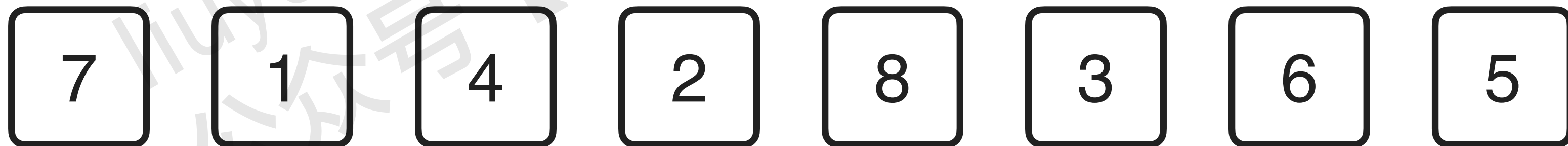


逆序数对个数

最朴素的想法：遍历所有的数据对

```
for(int i = 0; i < n; i++)  
    for(int j = i + 1; j < n; j++)  
        if(arr[i] > arr[j]) res ++;
```

$O(n^2)$



实验：暴力求解逆序数对问题

<https://leetcode-cn.com/problems/shu-zu-zhong-de-ni-xu-dui-lcof/>

使用归并排序法求解逆序数对个数问题

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

归并过程



归并过程

在整个区间，1 不和任何数字形成逆序数对



归并过程



归并过程



归并过程

2 和 3, 5, 6 形成逆序数对



归并过程



归并过程



归并过程

3 不和任何数字形成逆序数对



归并过程



归并过程

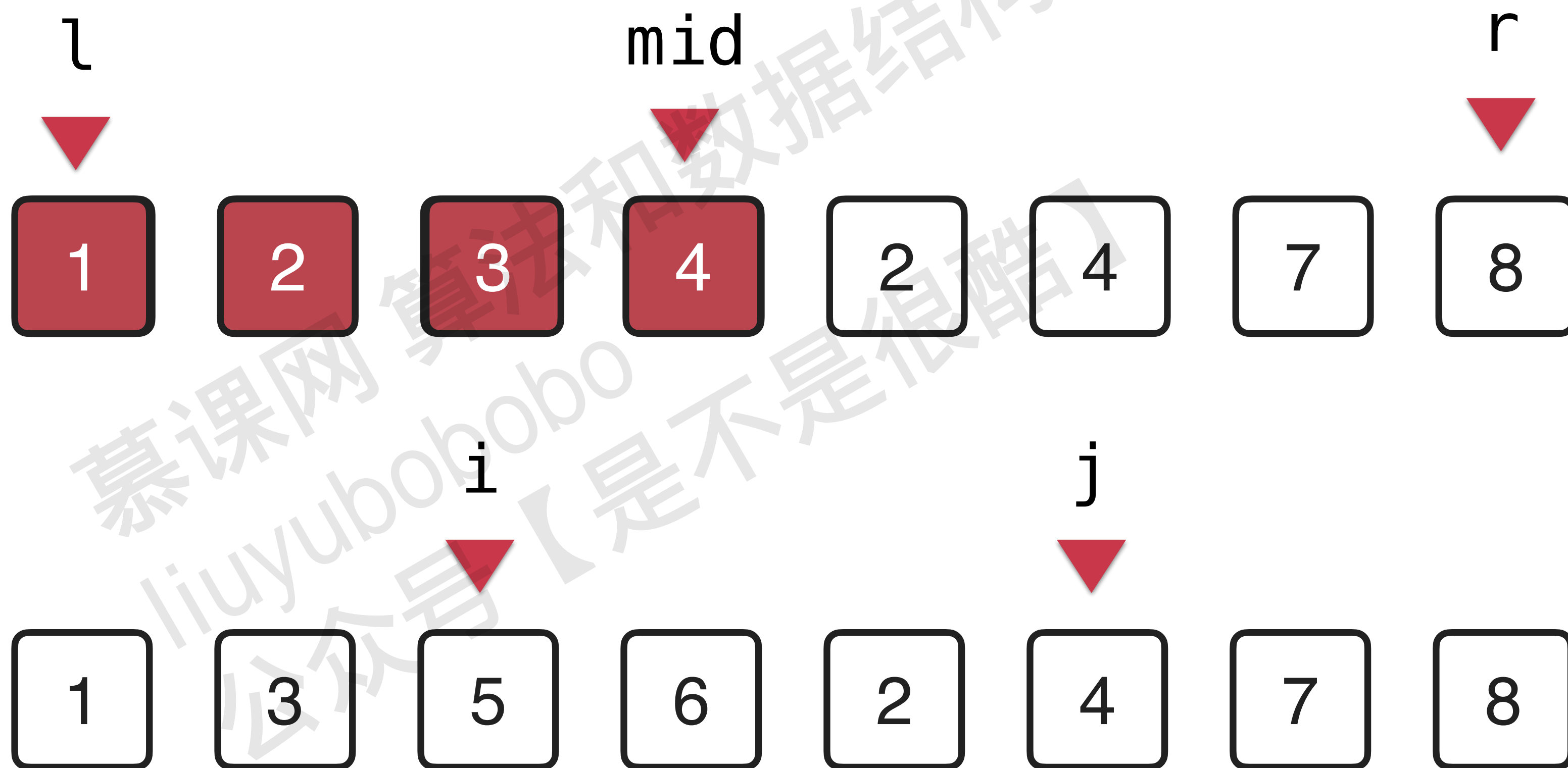


归并过程

4 和 5, 6 形成逆序数对



归并过程



归并过程

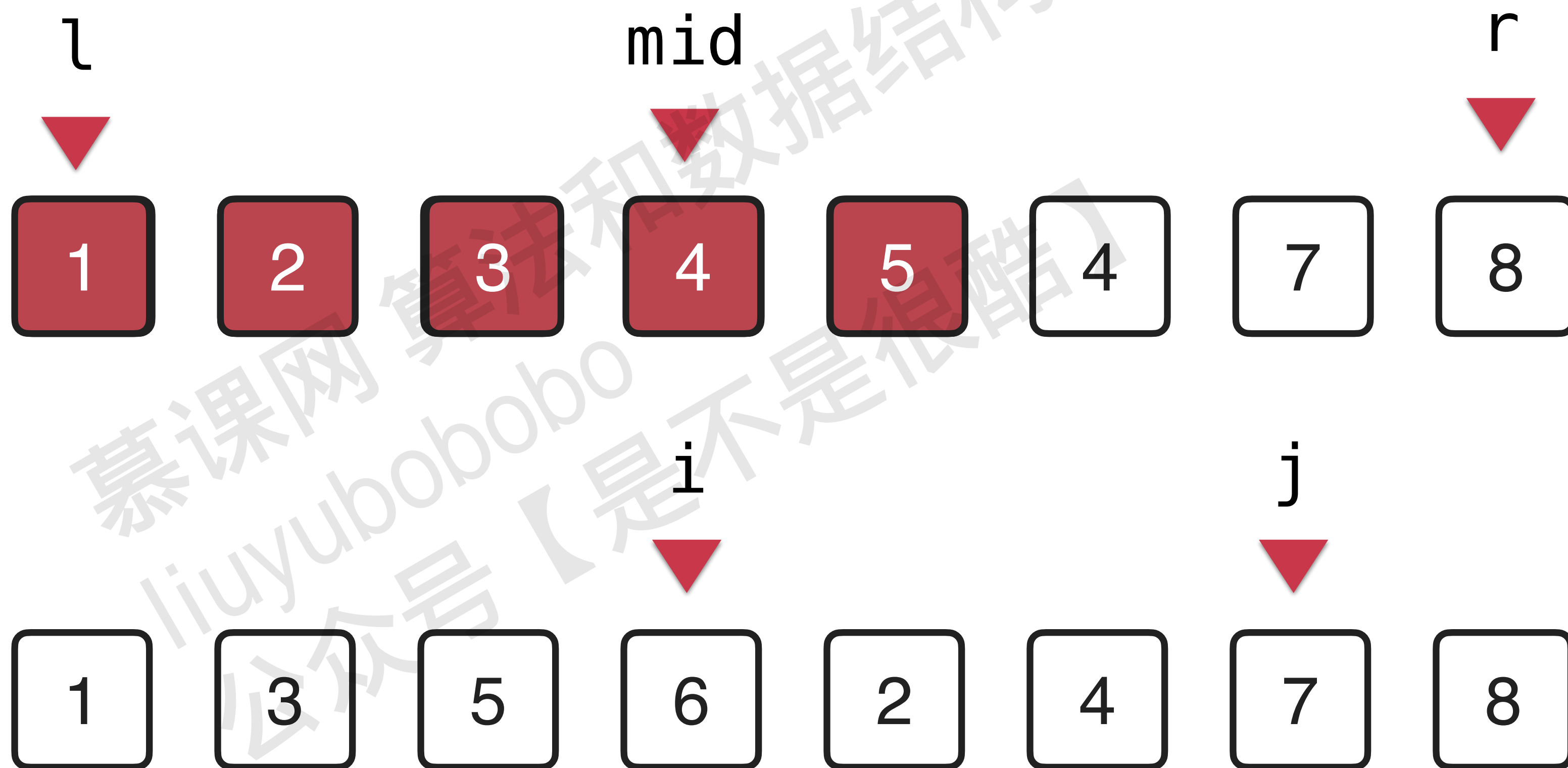


归并过程

5 不和任何数字形成逆序数对

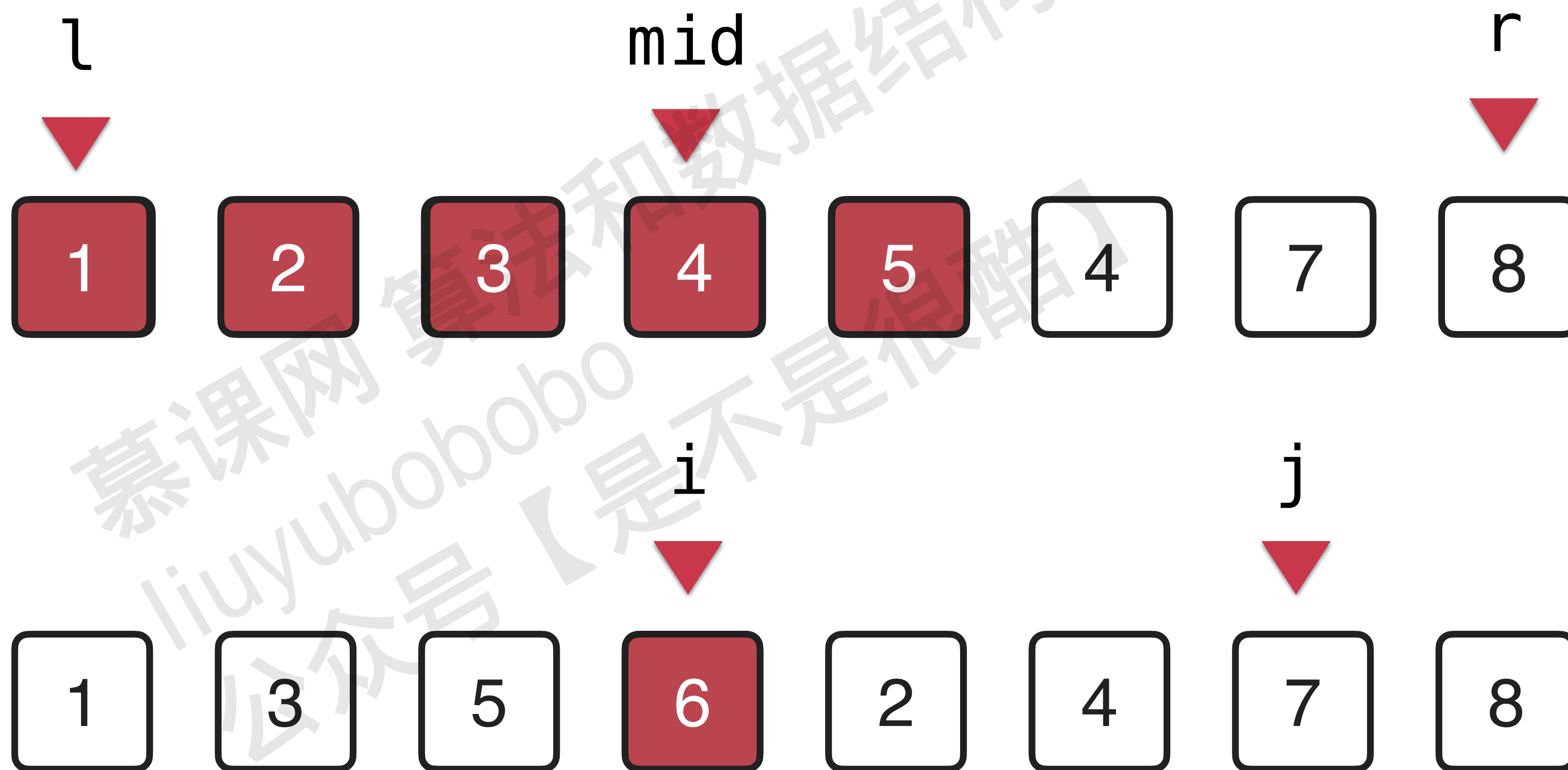


归并过程



归并过程

6 不和任何数字形成逆序数对



归并过程

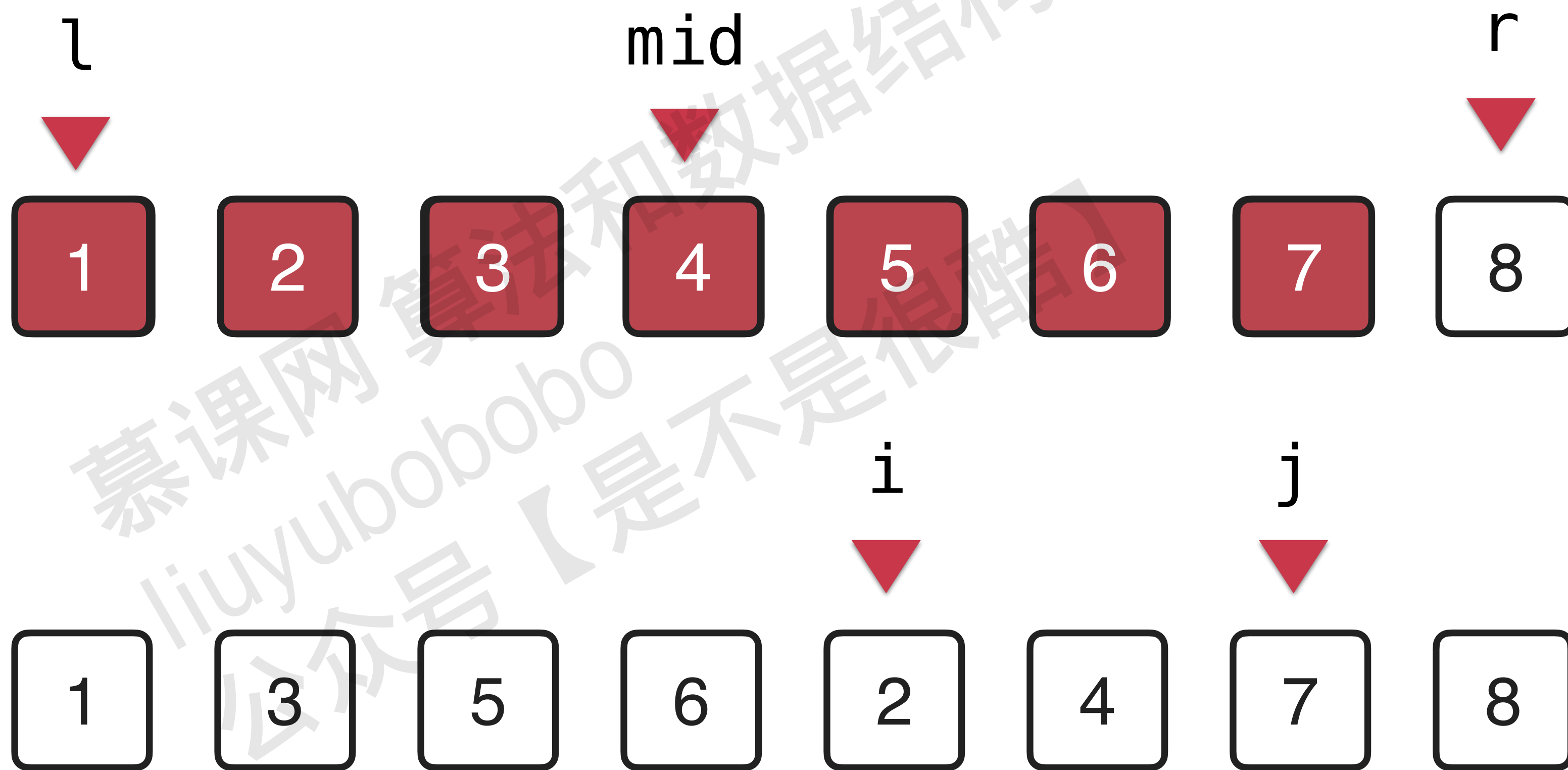


归并过程

7 不和任何数字形成逆序数对



归并过程



归并过程



归并过程

8 不和任何数字形成逆序数对

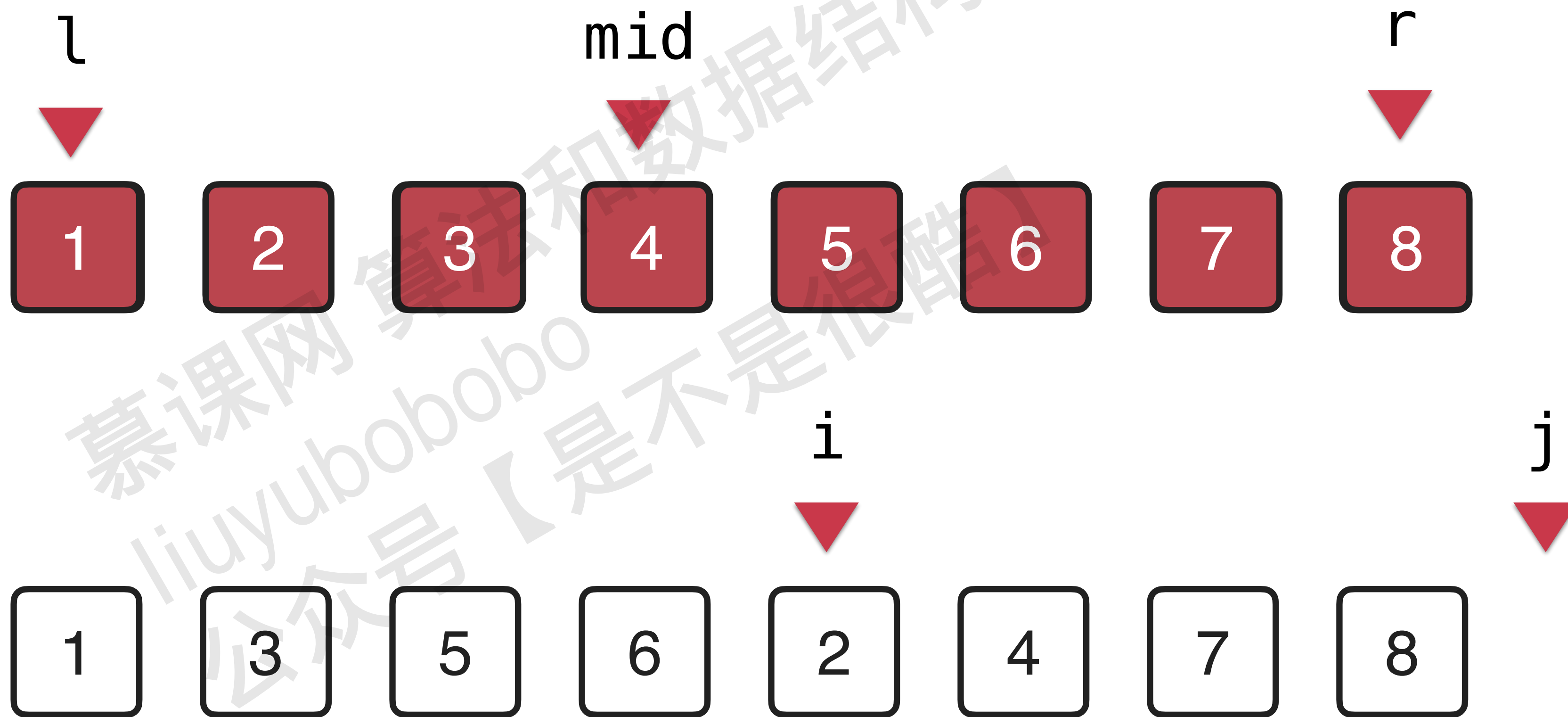


归并过程



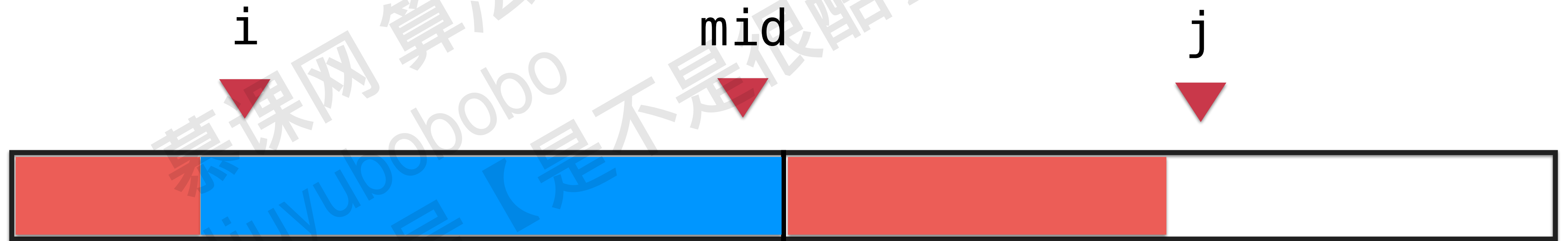
归并过程

后面区间的元素归并上来时，和前面区间剩余元素形成逆序数对



使用归并排序法求解逆序数对个数问题

后面区间的元素归并上来时，和前面区间剩余元素形成逆序数对



如果 $arr[j] < arr[i]$ $res += (mid - i + 1)$

实践：使用归并排序法求解逆序数对个数问题

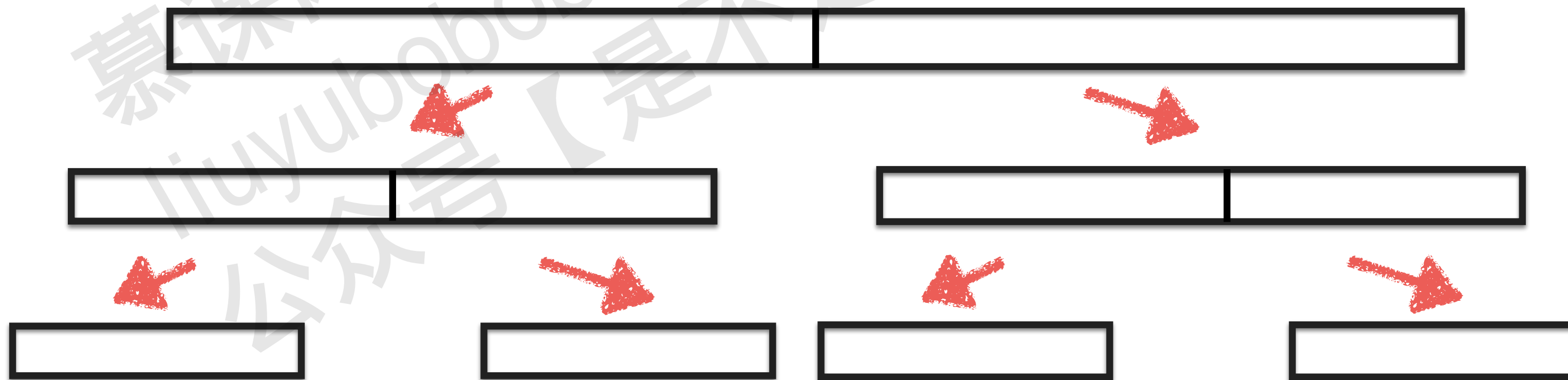
归并排序法总结

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

归并排序法

递归算法

分治算法



归并排序法

时间复杂度分析

$O(n \log n)$

加入对 merge 的优化，
对有序数组，是 $O(n)$ 的

归并排序法

优化 1：判断是否需要 merge

优化 2：对小规模数据使用插入排序

优化 3：只创建一个临时空间

归并排序算法不是原地排序算法

空间复杂度 $O(n)$

归并排序法

自顶向下的归并排序

自底向上的归并排序

归并排序法

解决逆序对数问题

使用自底向上的归并排序？

使用插入排序法优化？

归并排序法

慕课网 算法和数据结构 体系课程
liuyubobobo
公众号【是不是很有趣】

其他

欢迎大家关注我的个人公众号：是不是很酷



算法与数据结构体系课程

liuyubobobo