

慕课网《看得见的算法》

# 看得见的算法

讲师：liuyubobobo

版权所有，侵权必究

liuyubobobo

慕课网《看得见的算法》

# 排序算法可视化

讲师：lilyubobobo

版权所有，侵权必究

# 排序算法可视化

对 选择排序； 插入排序； 归并排序； 快速排序； 堆排序 进行可视化

# 排序算法可视化

## 算法与数据结构

难度 中级

时长 13小时35分钟

学习人数 2588

10.0分 ★★★★★

¥ 166.00

进入课程

课程介绍

组合套餐

课程章节

课程咨询 44

用户评价 226

# 排序算法可视化

本课程将不带领大家从0开始书写排序算法

对排序算法不感兴趣的同学可以跳过本章

以排序算法的可视化为主

通过可视化排序算法的过程， 深入理解排序算法

# 算法与数据结构

<https://github.com/liuyubobobo/Play-with-Algorithms/>



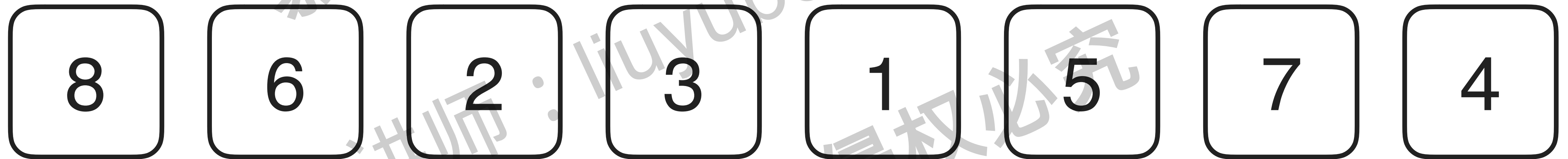
慕课网《看得见的算法》

# 选择排序可视化

讲师：lilyakobobo

版权所有，侵权必究

# 选择排序 Selection Sort

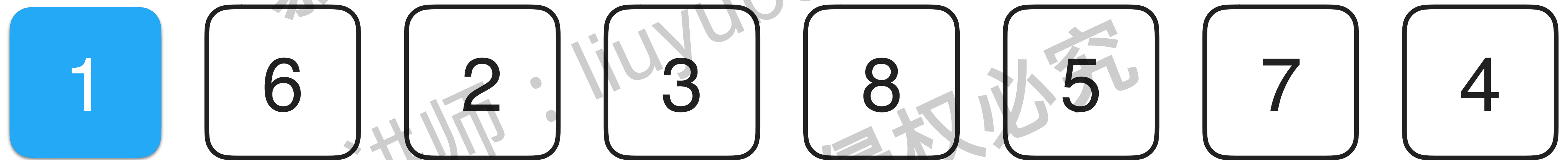




# 选择排序 Selection Sort



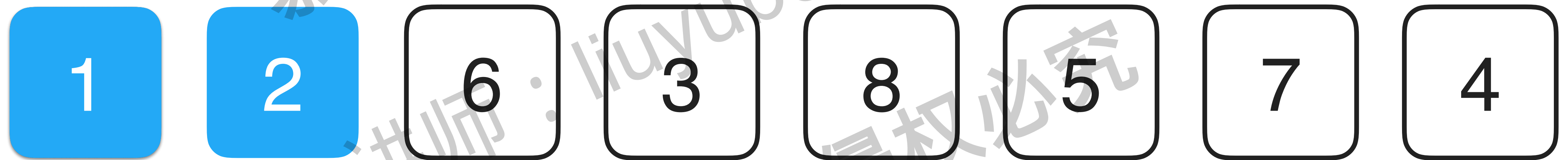
# 选择排序 Selection Sort



# 选择排序 Selection Sort



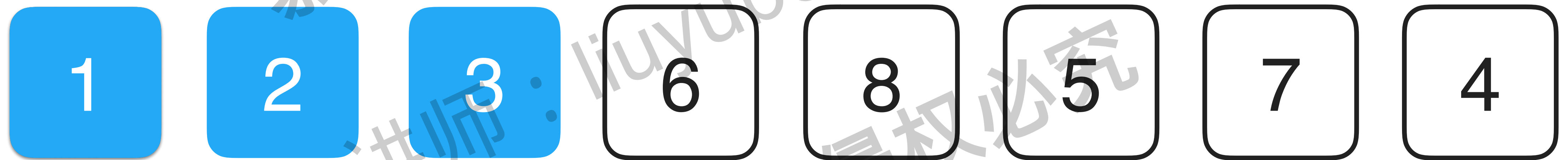
# 选择排序 Selection Sort



# 选择排序 Selection Sort

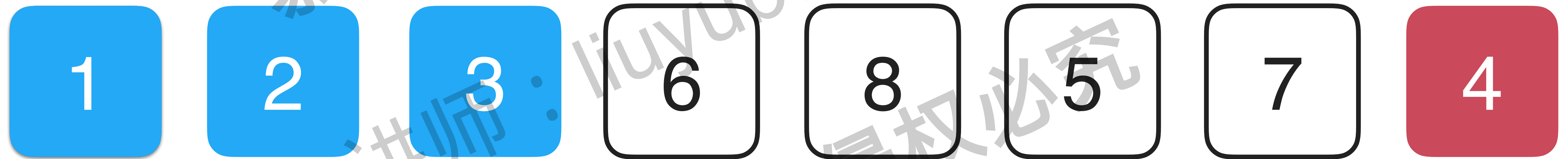


# 选择排序 Selection Sort





# 选择排序 Selection Sort



# 选择排序 Selection Sort



# 选择排序 Selection Sort



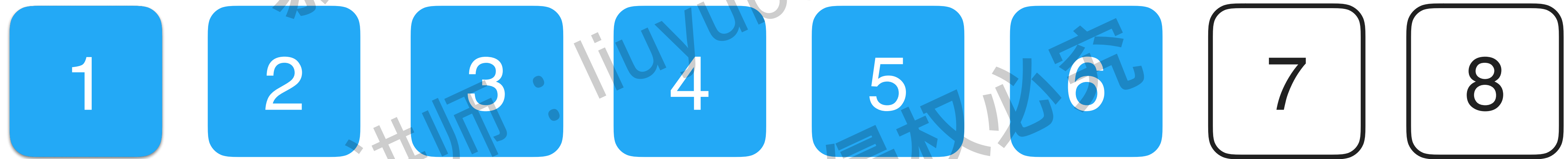
# 选择排序 Selection Sort



# 选择排序 Selection Sort



# 选择排序 Selection Sort





# 选择排序 Selection Sort



# 选择排序 Selection Sort



# 选择排序 Selection Sort



# 选择排序 Selection Sort



# 选择排序 Selection Sort

演示：选择排序可视化

慕课网《看得见的算法》

增加更多可视化效果

讲师：liuyubobobo

版权所有，侵权必究



# 增加更多可视化效果

演示：为选择排序可视化增加更多效果

# 选择排序 Selection Sort

交换最少的排序

慕课网《看得见的算法》

# 插入排序可视化

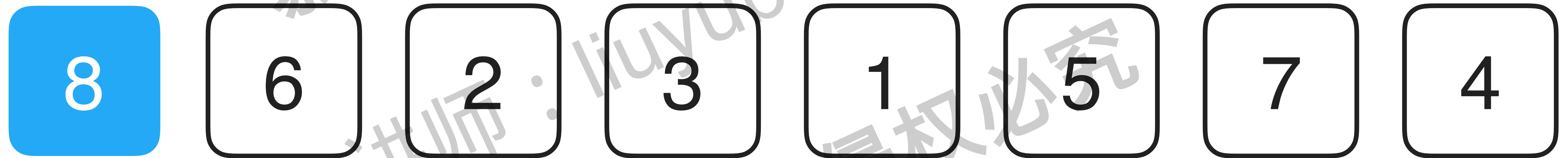
讲师：lilyakobobo

版权所有，侵权必究

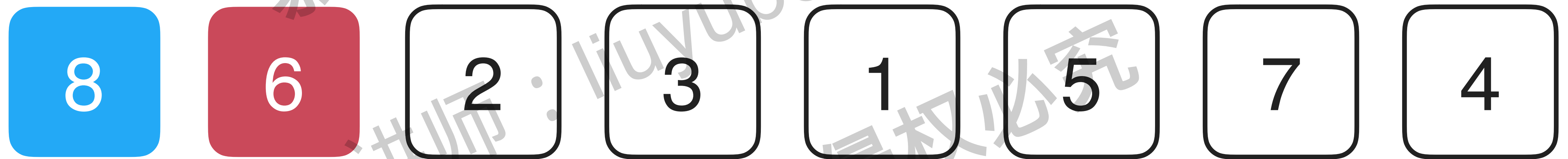
# 插入排序 Insertion Sort



# 插入排序 Insertion Sort

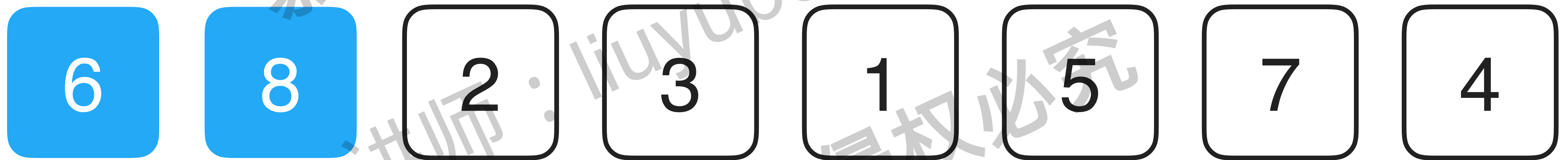


# 插入排序 Insertion Sort

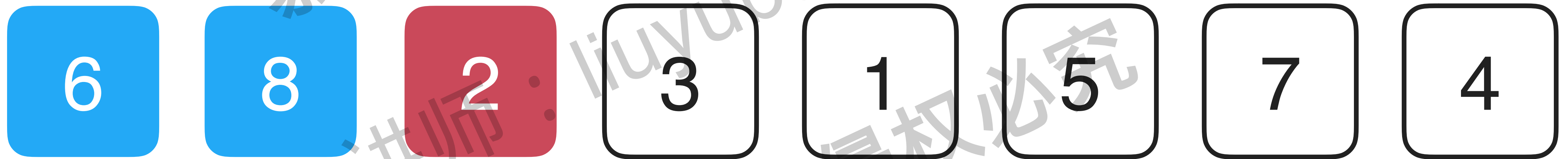




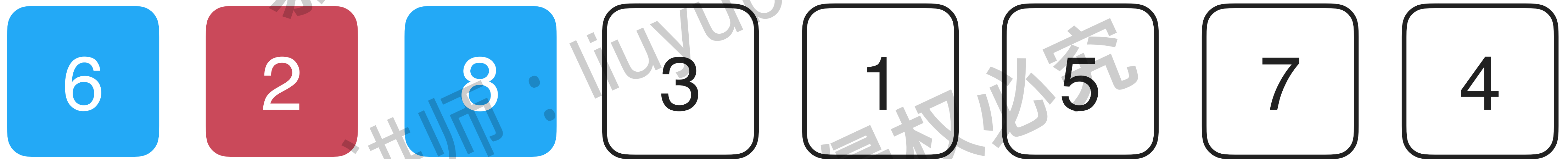
# 插入排序 Insertion Sort



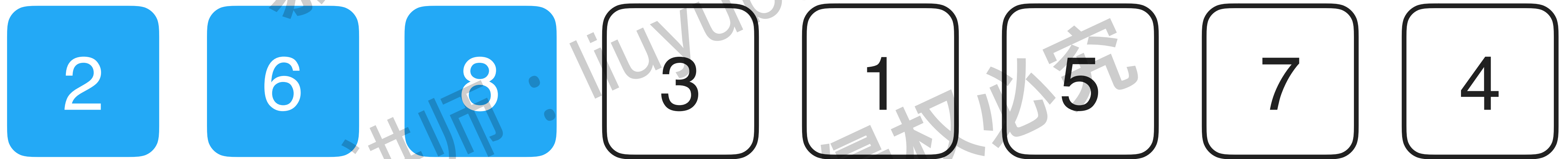
# 插入排序 Insertion Sort



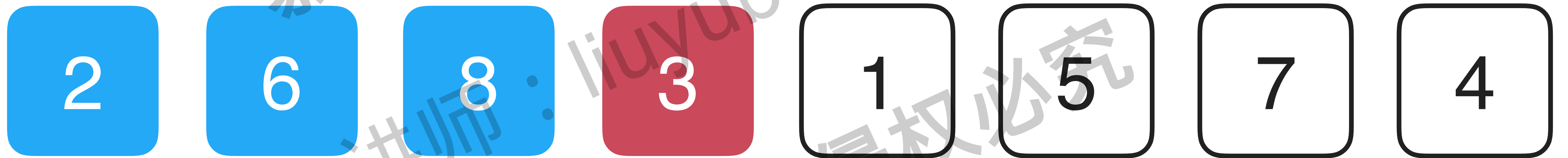
# 插入排序 Insertion Sort



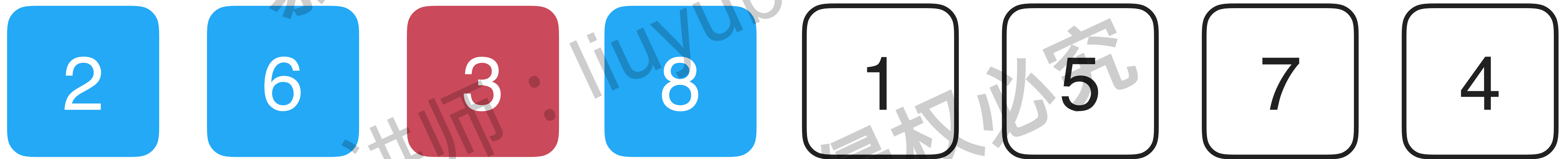
# 插入排序 Insertion Sort



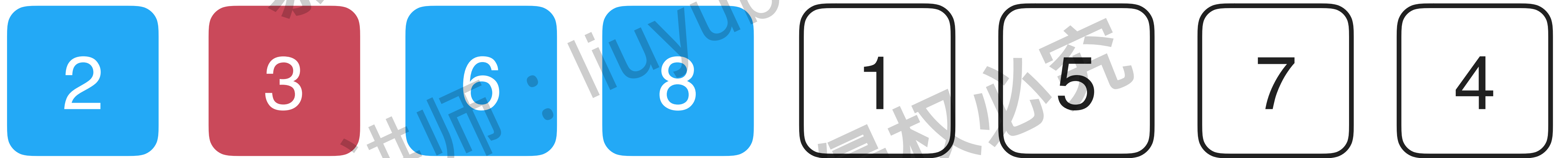
# 插入排序 Insertion Sort



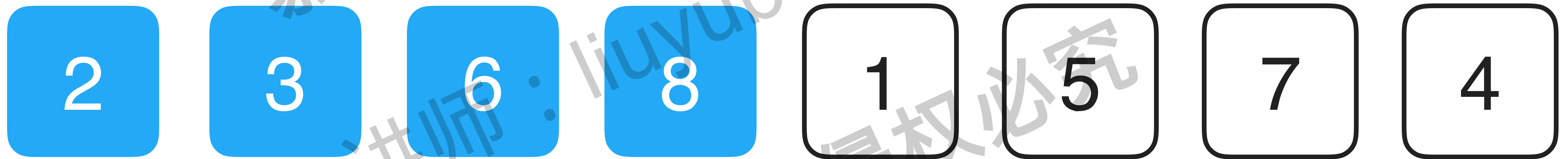
# 插入排序 Insertion Sort



# 插入排序 Insertion Sort



# 插入排序 Insertion Sort





# 插入排序 Insertion Sort

演示：插入排序可视化

慕课网《看得见的算法》

# 关于插入排序的其他

讲师：iluyubobobo

版权所有，侵权必究

# 插入排序 Insertion Sort

插入排序的优化：不交换数据，而使用数据移动的方式。

# 插入排序的特性

当数组近乎有序的时候，插入排序算法“进化”为 $O(n)$ 的算法

# 插入排序 Insertion Sort



# 插入排序 Insertion Sort



# 插入排序的特性

演示：插入排序面对近乎有序的数组

# 插入排序的特性

对于近乎有序的数组，插入排序算法“进化”为 $O(n)$ 的算法。

同样是 $O(n^2)$ 的算法，但是 $2n^2$ 和 $12n^2$ 差距很大。

在 $n$ 比较小的时候，插入排序比 $O(n \log n)$ 的排序算法有优势。

插入排序法经常用作是高级排序算法在处理到小样本时的一个优化。



# 可否在插入排序中使用二分查找？



可以。

# 可否在插入排序中使用二分查找？



版权所有，侵权必究

# 可否在插入排序中使用二分查找？



e

# 可否在插入排序中使用二分查找？



e

# 可否在插入排序中使用二分查找？



# 可否在插入排序中使用二分查找？

可以。

但仍然无法改变插入排序是 $O(n^2)$ 的算法。

此时，面对近乎有序的数组，无法“进化”为 $O(n)$ 的算法。

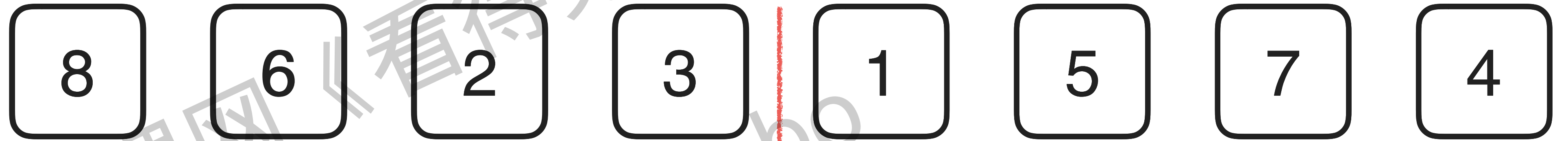
慕课网《看得见的算法》

# 归并排序法

讲师：liuyubobobo

版权所有，侵权必究

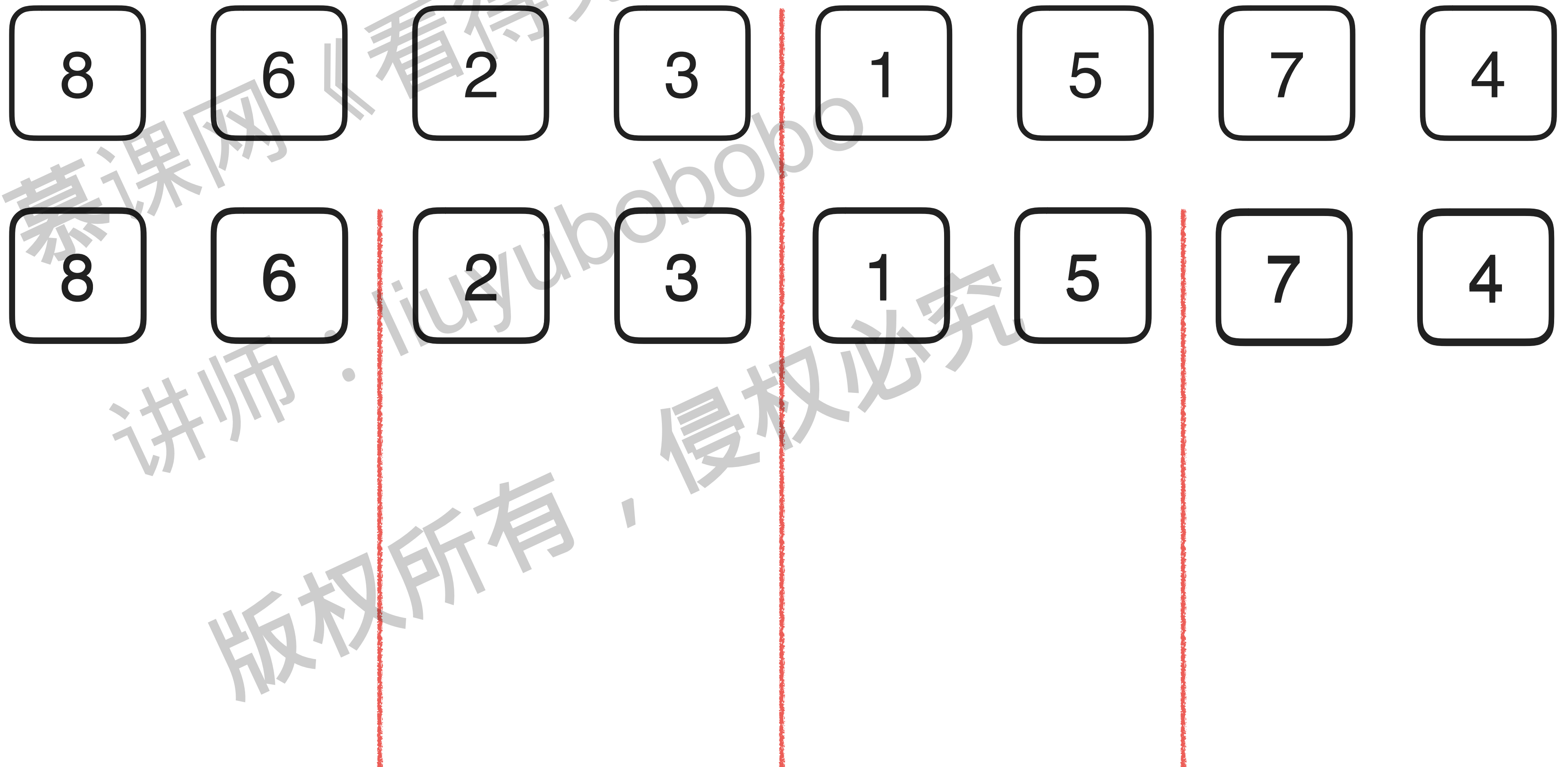
# 归并排序 Merge Sort



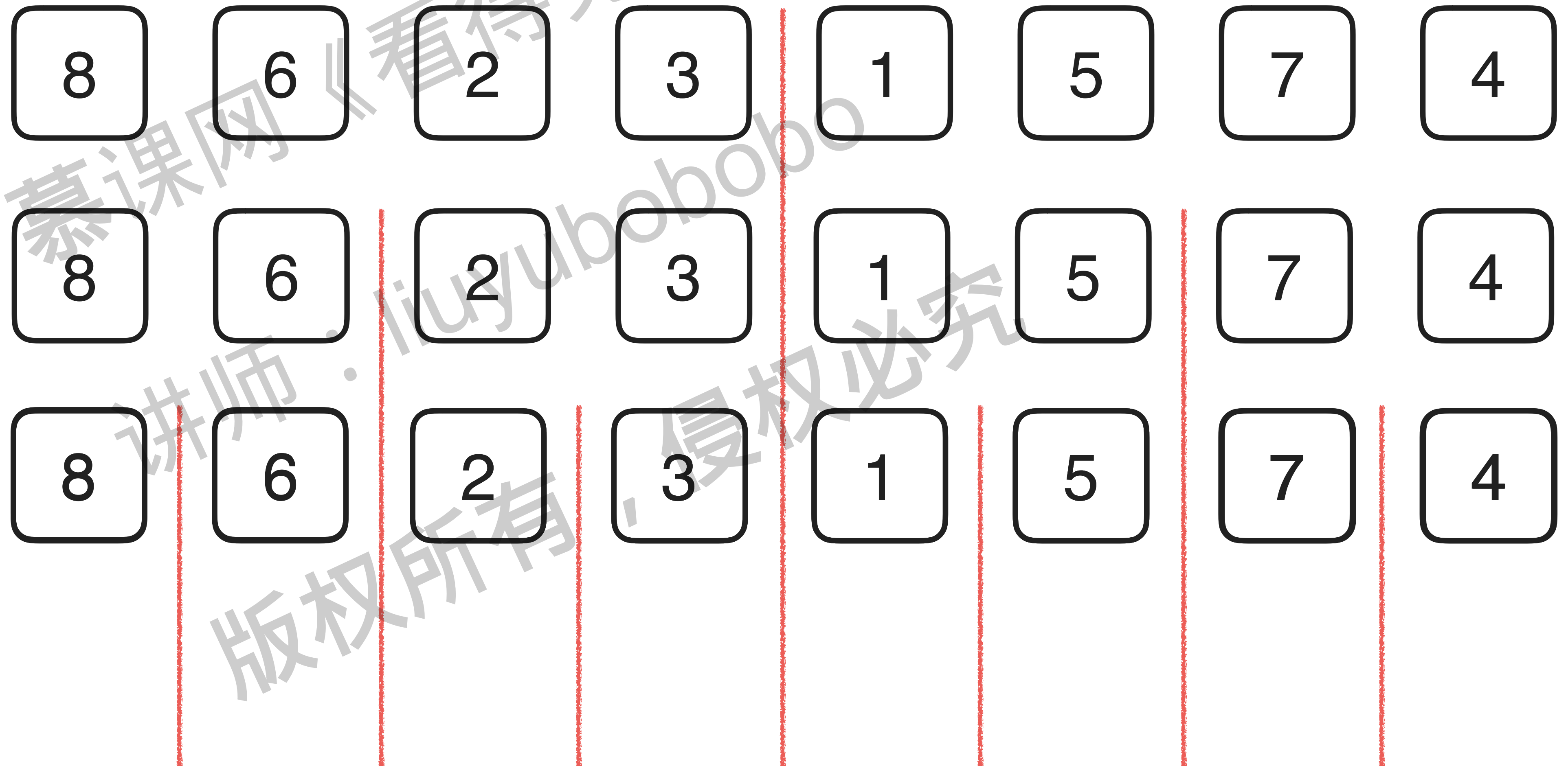
慕课网《看得见的算法》  
讲师：liuyubobobo  
版权所有，侵权必究



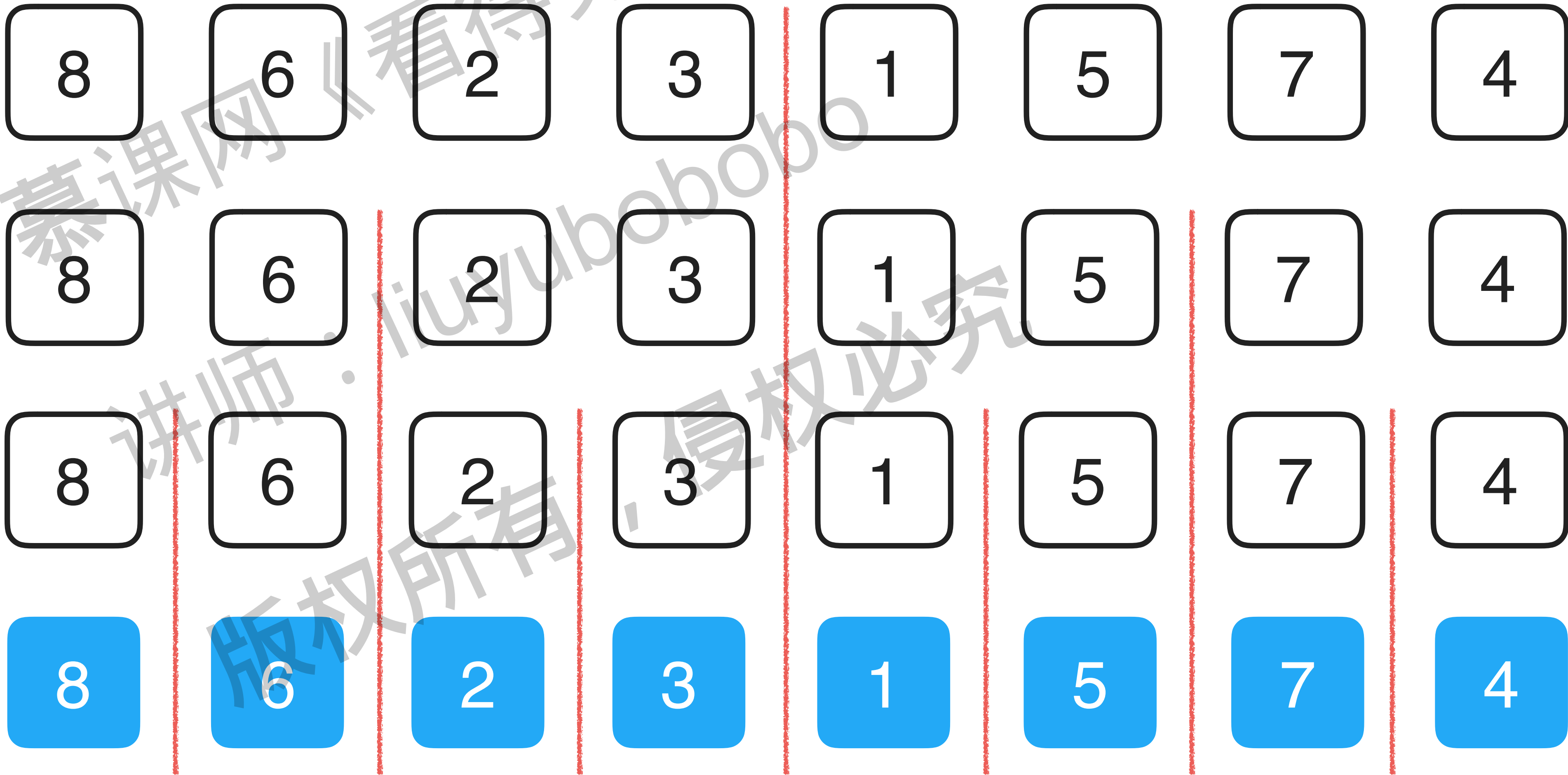
# 归并排序 Merge Sort



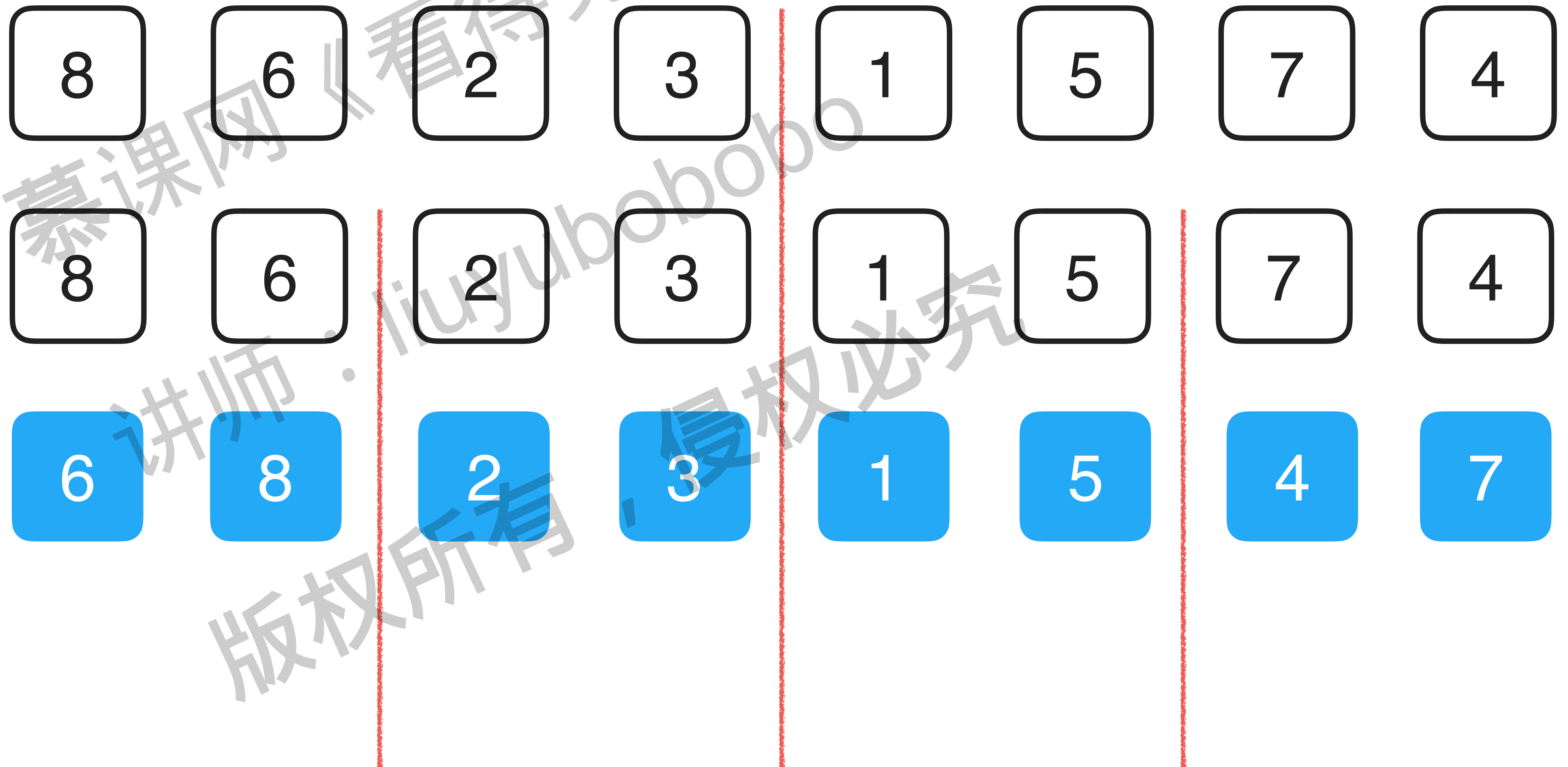
# 归并排序 Merge Sort



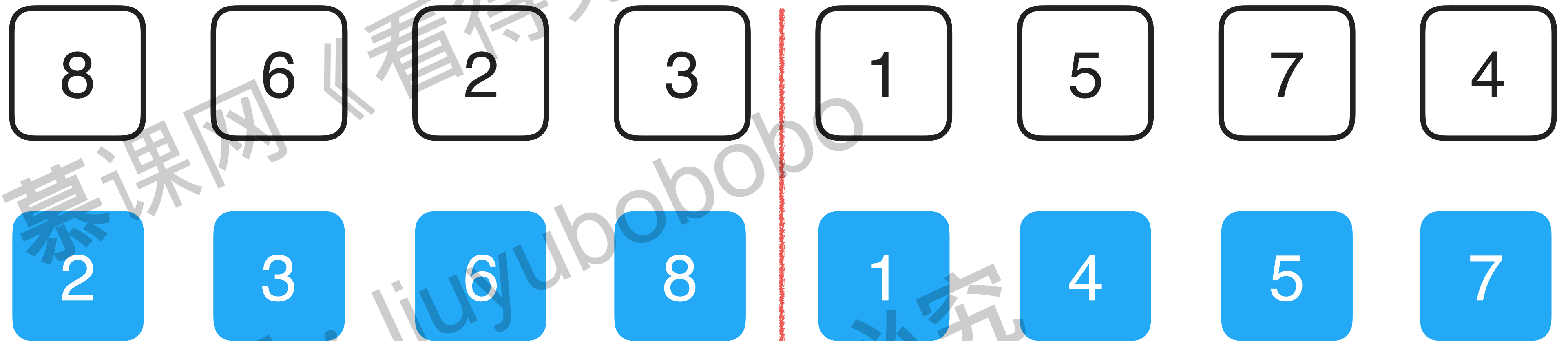
# 归并排序 Merge Sort



# 归并排序 Merge Sort



# 归并排序 Merge Sort

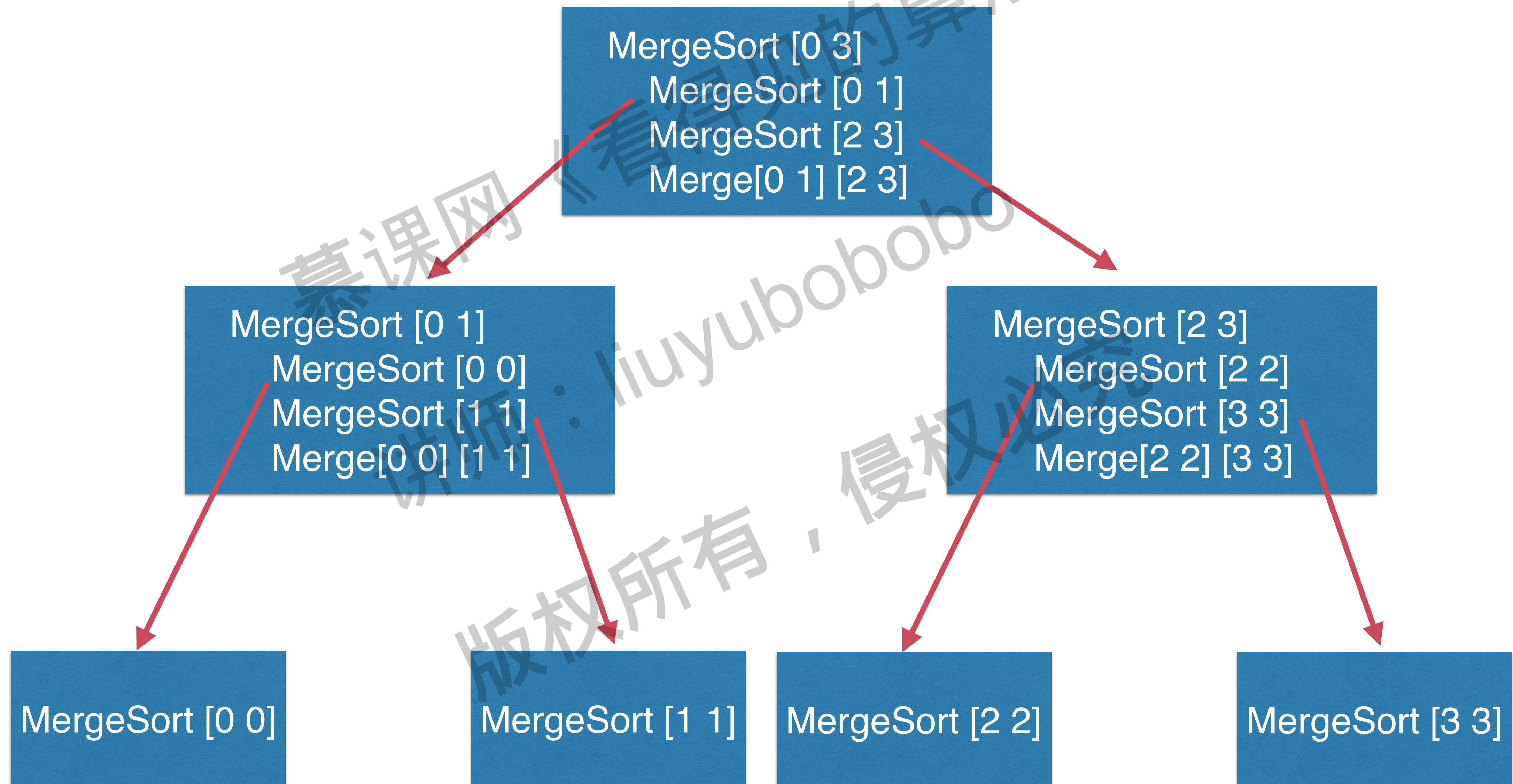


# 归并排序 Merge Sort

演示：归并排序算法代码



# 归并排序 Merge Sort



# 归并排序 Merge Sort

演示：归并排序算法中添加输出查看结果



# 归并排序 Merge Sort

递归和普通函数调用没有区别

普通函数调用，A中调用B，B结束以后继续执行A

递归函数调用，A中调用A，下面的A结束以后继续执行上面的A

递归是要返回的 - 回溯

# 归并排序 Merge Sort

演示：归并排序优化的代码呈现

慕课网《看得见的算法》

# 归并排序可视化

讲师：luyubobobo

版权所有，侵权必究

# 归并排序可视化

演示：归并排序可视化

# 归并排序可视化

演示：自底向上的归并排序的可视化

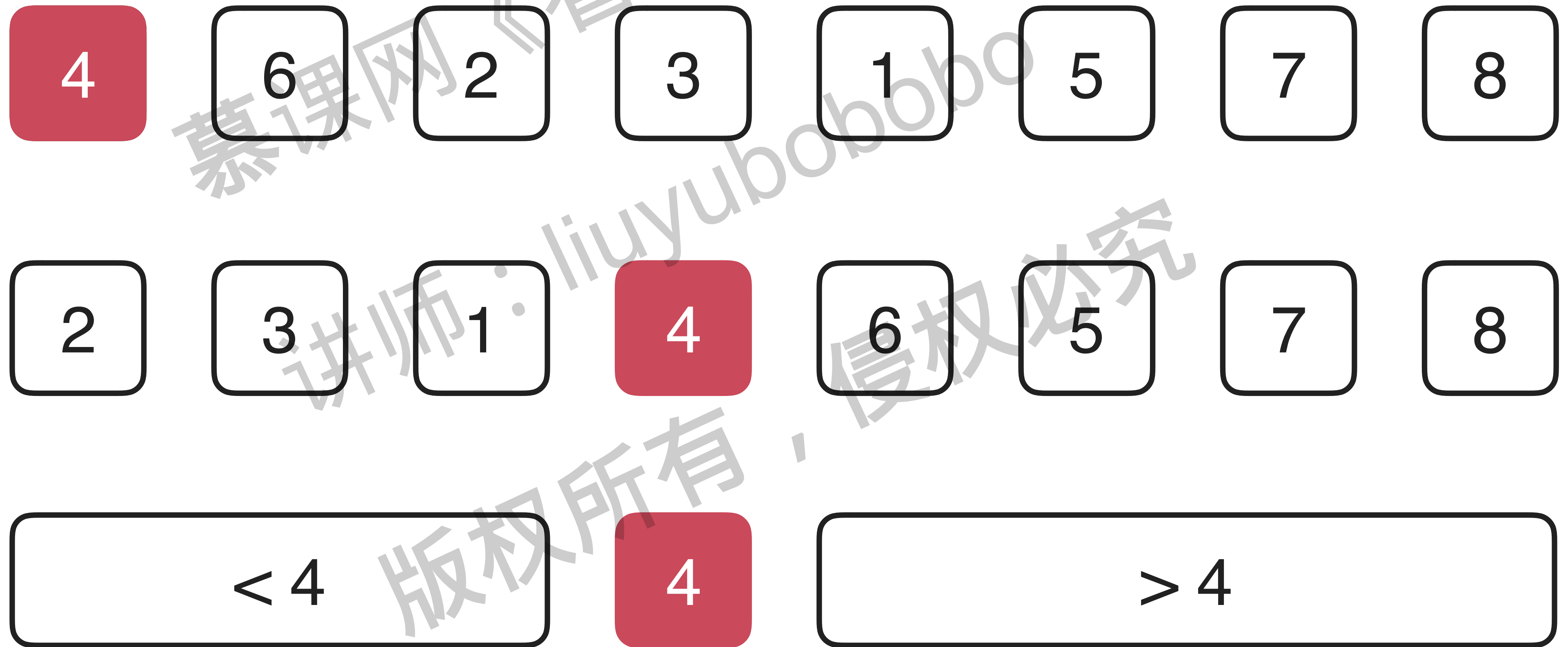
慕课网《看得见的算法》

# 快速排序的可视化

讲师：liuyubobobo

版权所有，侵权必究

# 快速排序 Quick Sort



# 快速排序 Quick Sort





# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



$l$

$j$

$i$

# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



$e > v$

# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



$l$

$j$

$i$

$e < v$

# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



$e < v$

# Partition

$\text{arr}[l+1 \dots j] < v$

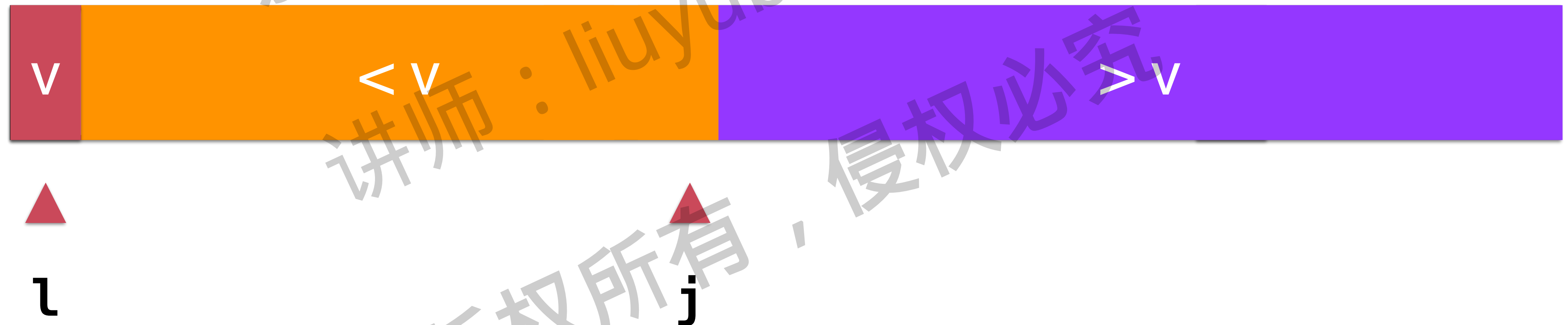
$\text{arr}[j+1 \dots i-1] > v$



# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



# 快速排序可视化

演示：快速排序可视化



# 带随机化的快速排序可视化

慕课网《看得见的算法》  
讲师：lucyubobobo

版权所有，侵权必究

# 快速排序可视化

演示：加入近乎有序的数组，演示不随机化的问题

# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



$l$

$j$

$i$

# Partition

$\text{arr}[j+1 \dots i-1] > v$



l

i

j

# Partition

$\text{arr}[j+1 \dots i-1] > v$



**l**



**j**



**i**

# Partition

$\text{arr}[j+1 \dots i-1] > v$



# Partition

$\text{arr}[j+1 \dots i-1] \geq v$



# 快速排序可视化

慕课网

《看得见的算法》

讲师：liuyubobobo

演示：添加随机化

版权所有，侵权必究



# 快速排序可视化

演示：对于相同元素的问题

# 快速排序可视化

演示：对于相同元素的数组的排序问题

# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] > v$



# Partition

$\text{arr}[l+1 \dots j] < v$

$\text{arr}[j+1 \dots i-1] \geq v$



# Partition

$\text{arr}[j+1 \dots i-1] \geq v$



# Partition

$\text{arr}[l+1 \dots j] \leq v$

$\text{arr}[j+1 \dots i-1] > v$



# Partition

$\text{arr}[l+1 \dots j] \leq v$



$l$



$j$



$i$



# 双路快速排序算法的可视化

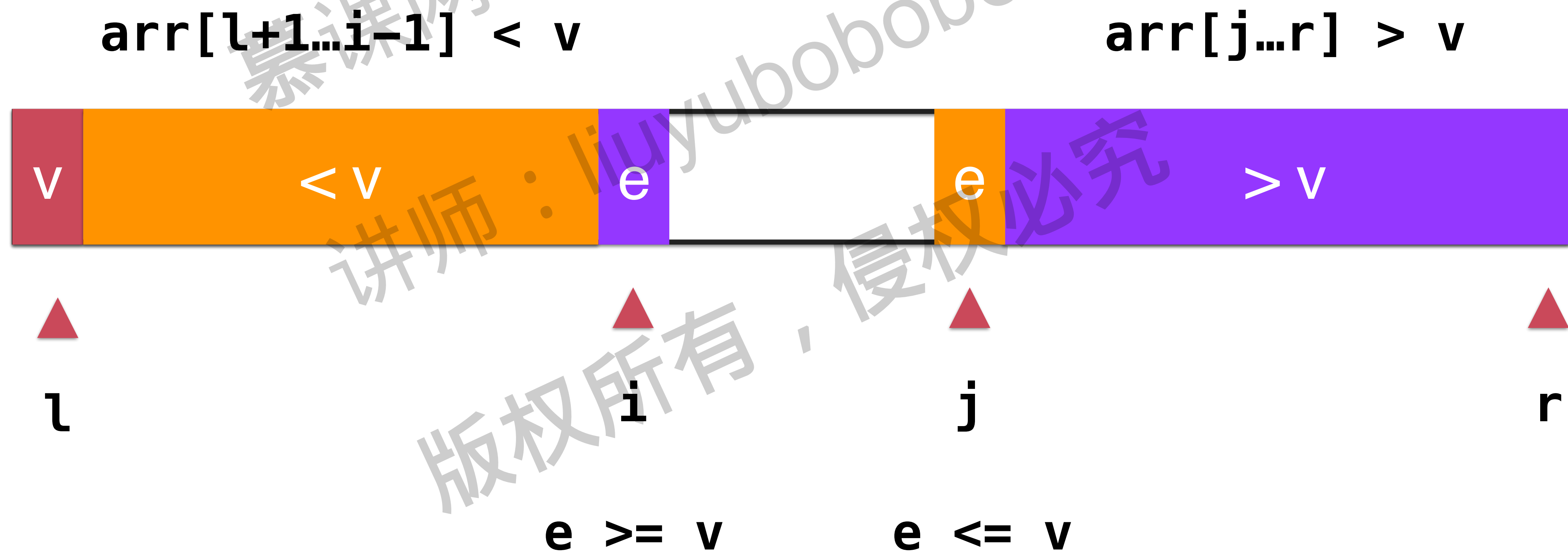
慕课网《看得见的算法》  
讲师：liuyubobobo

版权所有，侵权必究

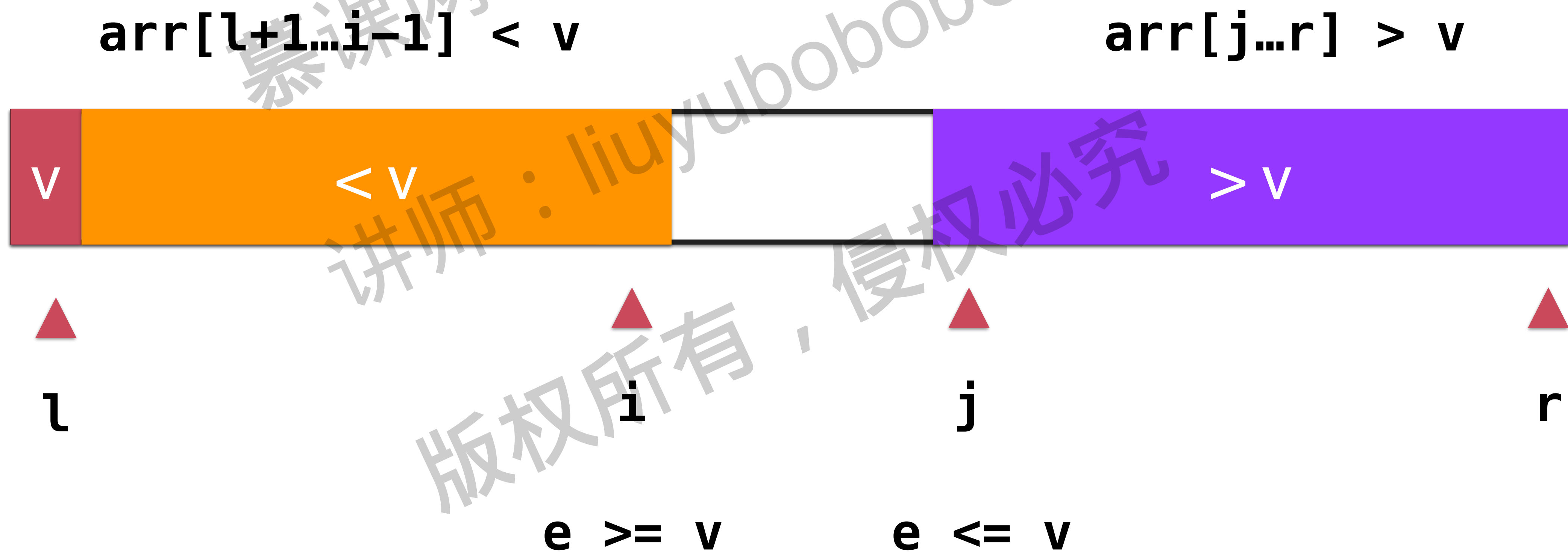
# 双路快速排序的Pa



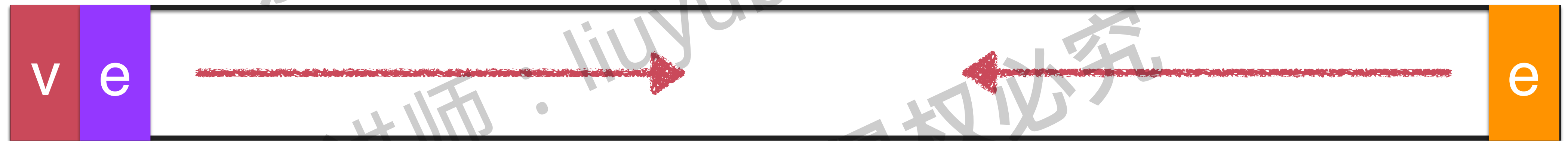
# 双路快速排序的Partition



# 双路快速排序的Partition



# 双路快速排序的Partition



$l$

$e \geq v$

$e \leq v$

# 双路快速排序可视化

演示：双路快速排序的可视化

# 三路快速排序算法的可视化

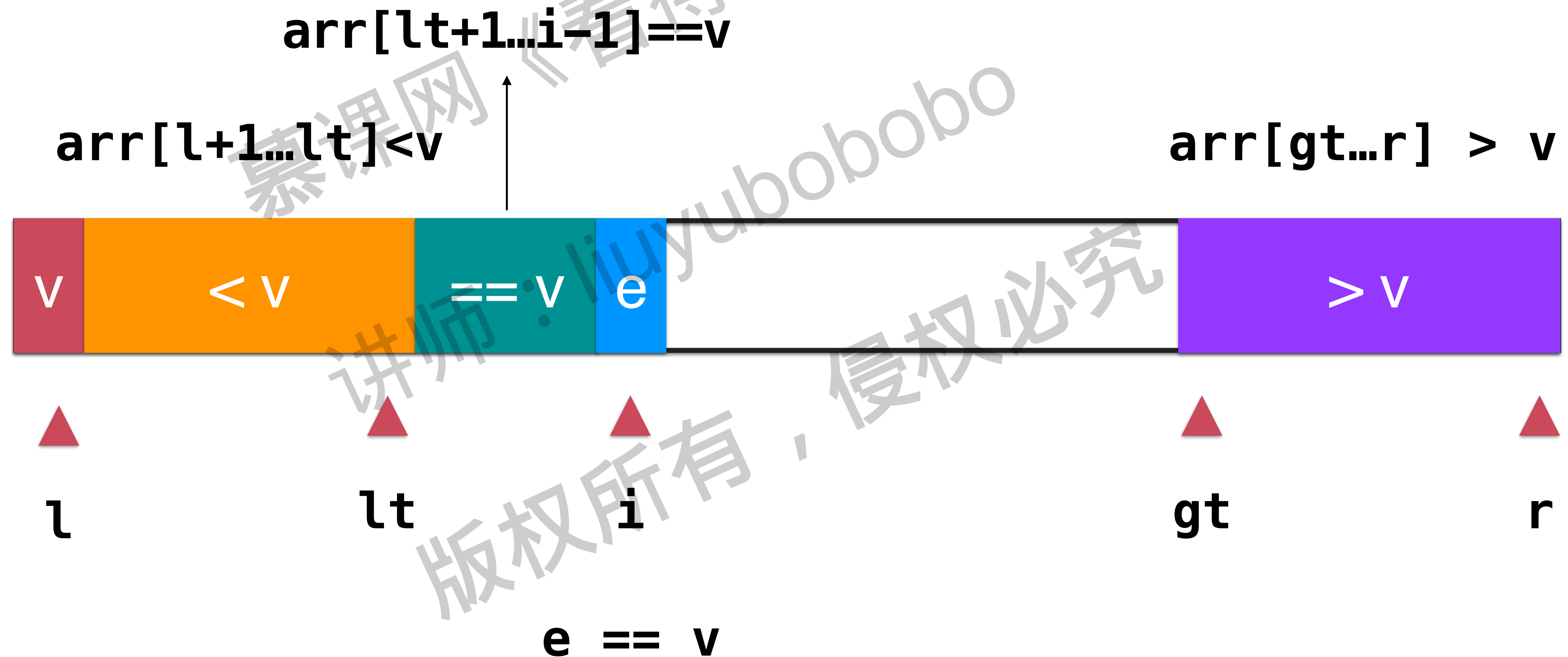
慕课网《看得见的算法》

讲师：liuyubobobo

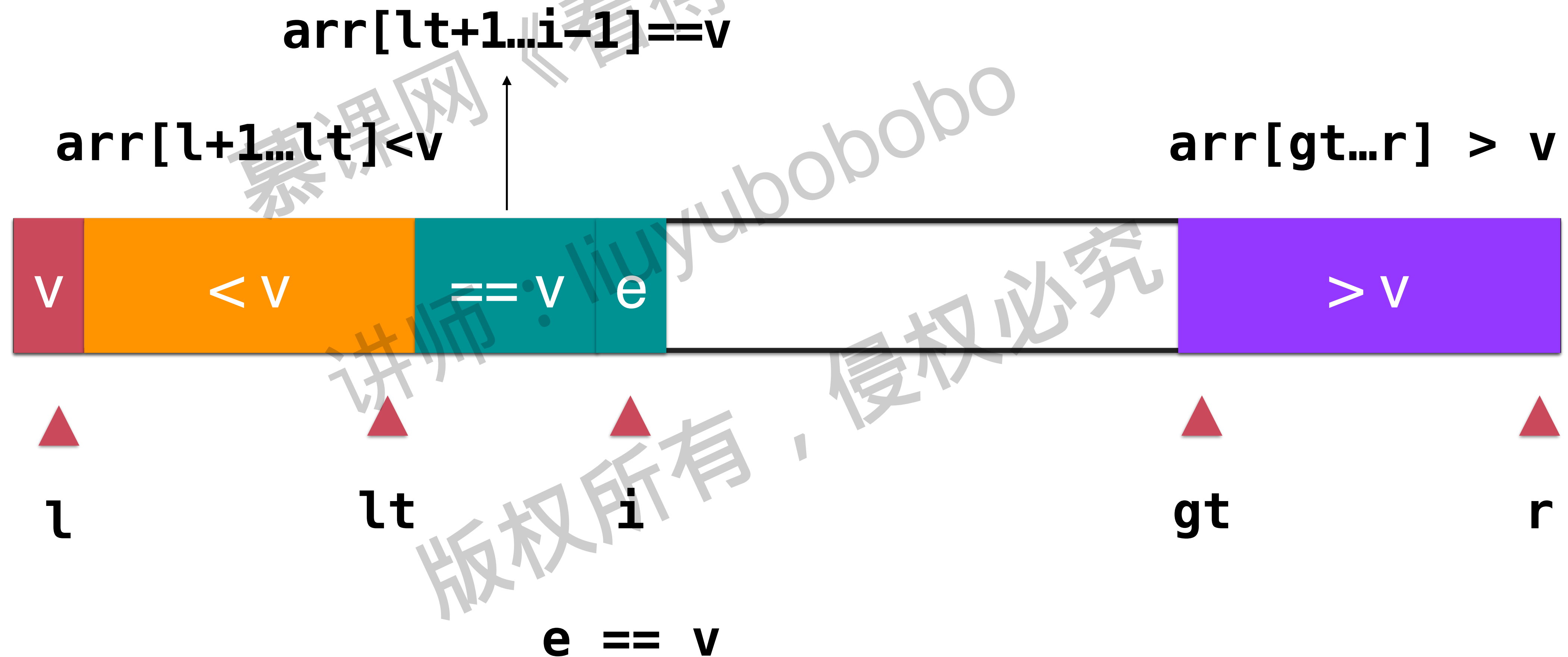
版权所有，侵权必究



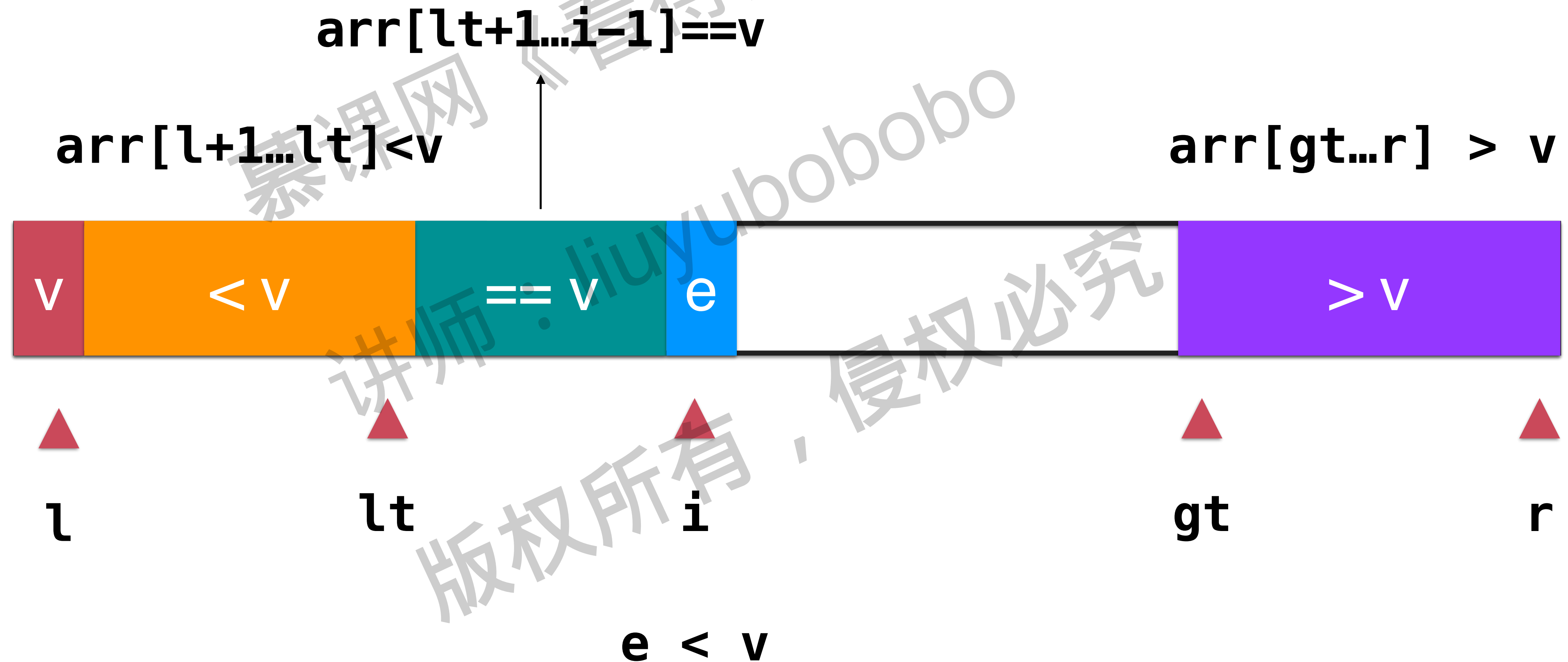
# Quick Sort 3 Ways



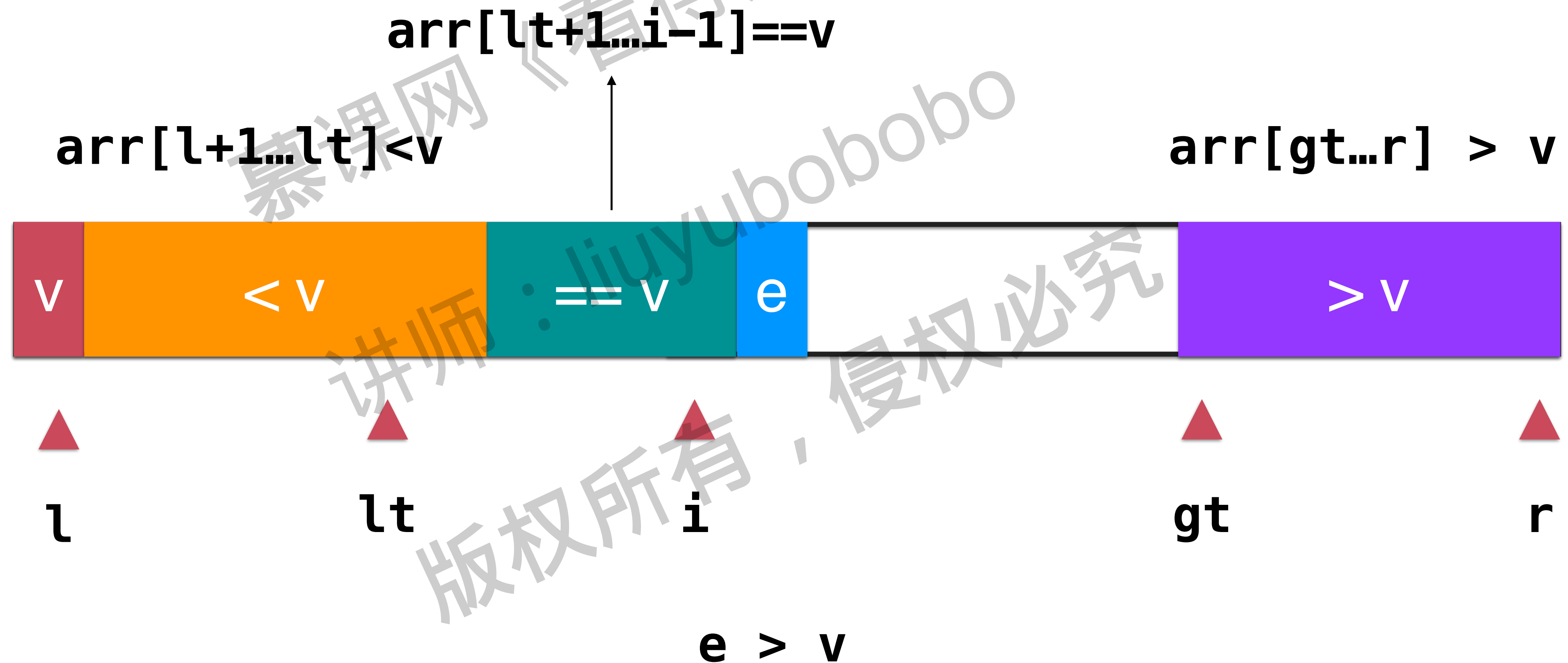
# Quick Sort 3 Ways



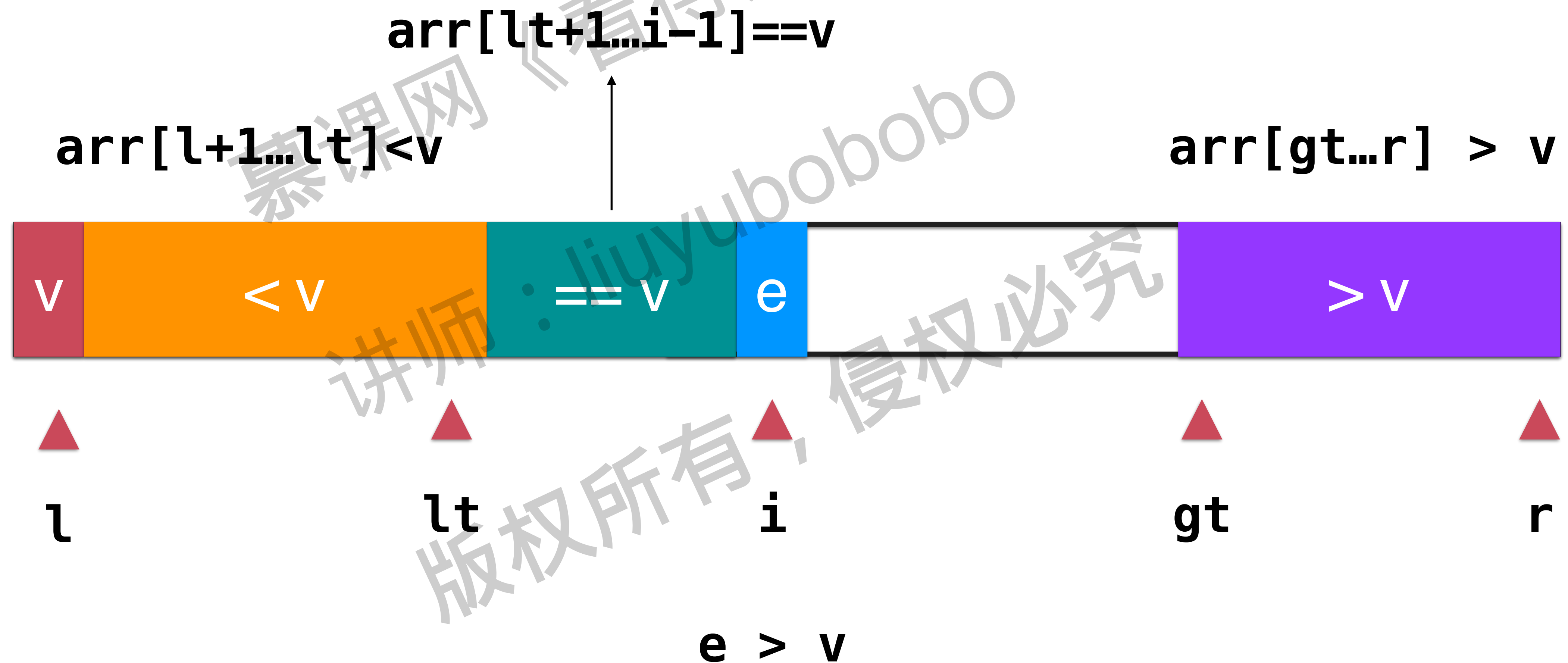
# Quick Sort 3 Ways



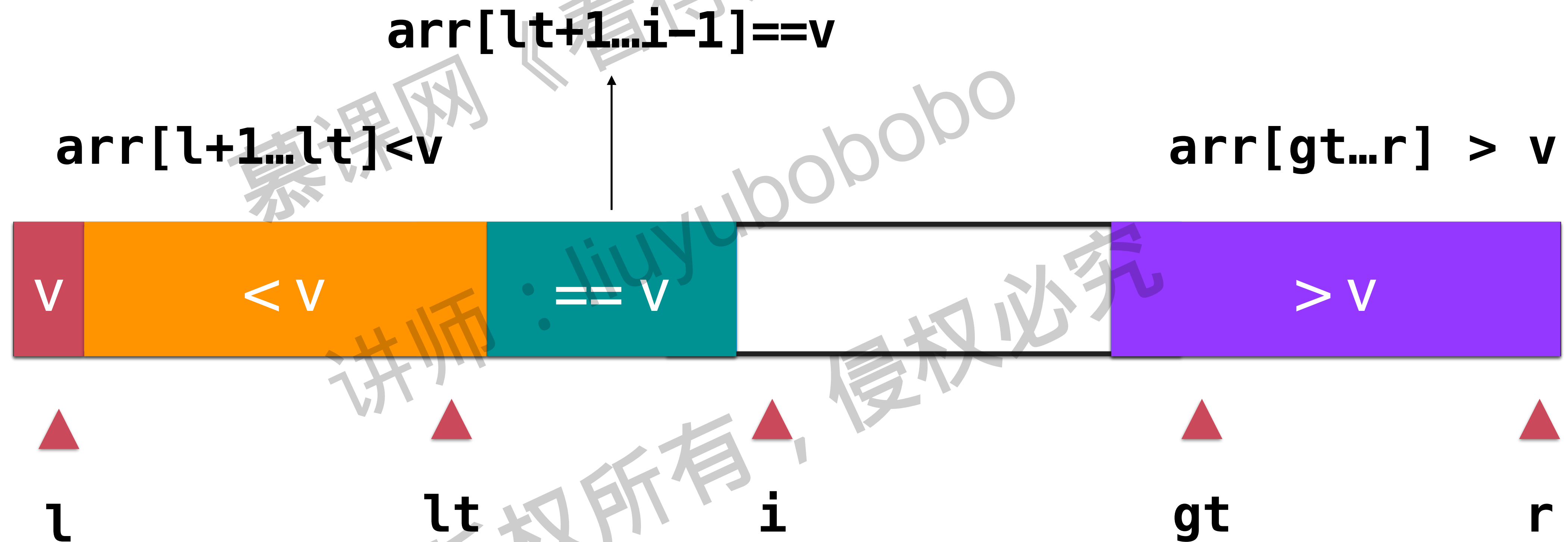
# Quick Sort 3 Ways



# Quick Sort 3 Ways



# Quick Sort 3 Ways



# Quick Sort 3 Ways

$\text{arr}[l+1 \dots lt] < v$

$\text{arr}[lt+1 \dots i-1] == v$

$\text{arr}[gt \dots r] > v$



$l$

$lt$

$gt$

$r$

$i$

# Quick Sort 3 Ways

$\text{arr}[l \dots lt-1] < v$

$\text{arr}[lt \dots gt-1] == v$

$\text{arr}[gt \dots r] > v$



$l$



$lt$



$gt$



$r$



# 三路快排 Quick Sort 3 Ways



# 三路快速排序可视化

演示：三路快速排序的可视化

慕课网《看得见的算法》

# 堆排序的可视化

讲师：liuyubobobo

版权所有，侵权必究

# 原地堆排序

max



v

Max Heap

# 原地堆排序

max



v

Max Heap

w

# 原地堆排序

max



# 原地堆排序

max



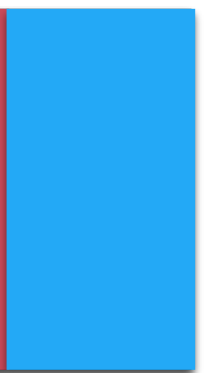
# 原地堆排序

max



v

Max Heap





# 原地堆排序

max



# 原地堆排序

max



# 原地堆排序



# 原地堆排序

Max Heap

# 原地堆排序

演示：原地堆排序算法的可视化

# 其他

欢迎大家关注我的个人公众号：是不是很酷



慕课网《看得见的算法》

# 看得见的算法

讲师：liuyubobobo

版权所有，侵权必究

liuyubobobo