

排序

快速排序：轴点

14-A7

邓俊辉

deng@tsinghua.edu.cn

左朱雀之芨芨兮，右苍龙之躍躍

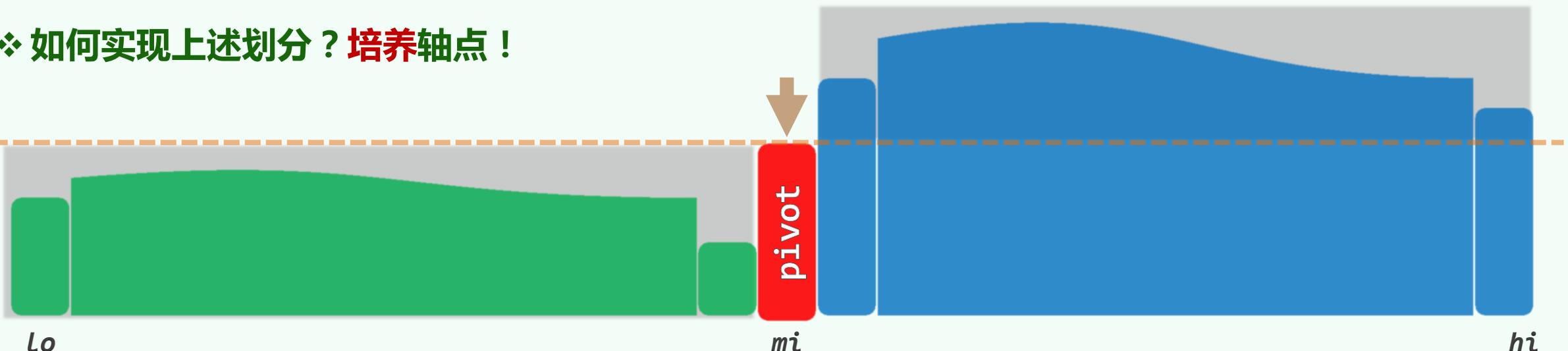
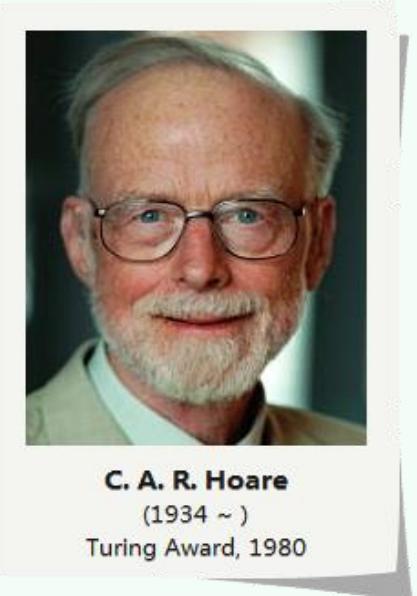
分而治之

- ❖ pivot : 左侧/右侧的元素，均不比它更大/更小
- ❖ 以轴点为界，自然划分 : $\max([0, mi]) \leq \min((mi, hi))$
- ❖ 前缀、后缀各自（递归）排序之后，原序列自然有序

`sorted(S) = sorted(SL) + sorted(SR)`

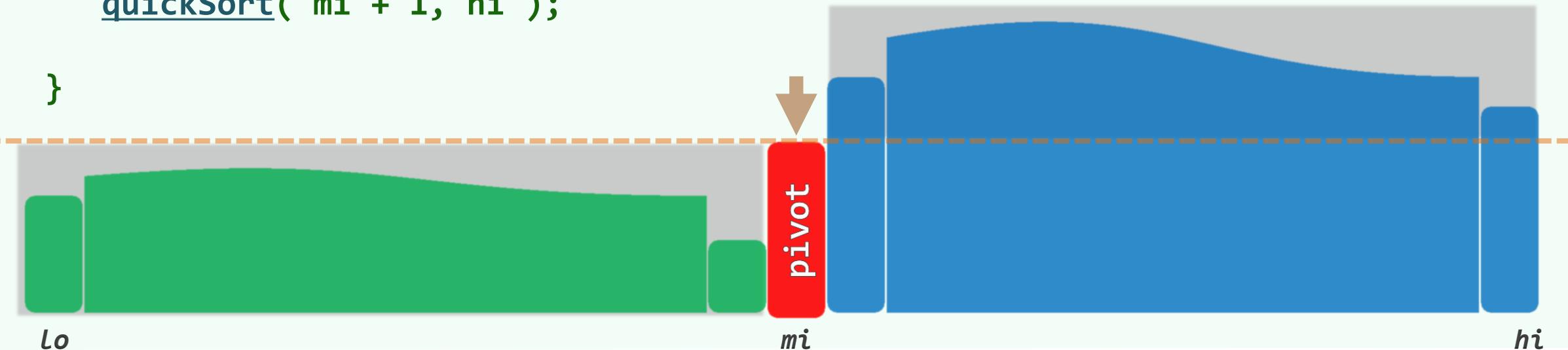
❖ mergesort的难点在于合，而quicksort在于分

❖ 如何实现上述划分？培养轴点！



快速排序

```
❖ template <typename T> void Vector<T>::quickSort( Rank lo, Rank hi ) {  
    if ( hi - lo < 2 ) return;  
  
    Rank mi = partition( lo, hi ); //能否足够高效?  
  
    quickSort( lo, mi );  
  
quickSort( mi + 1, hi );  
}
```



轴点

- ❖ 坏消息： 在原始序列中，轴点未必存在...
- ❖ 必要条件： 轴点必定已然就位 //尽管反之不然
- ❖ 特别地： 在有序序列中，所有元素皆为轴点
反之亦然
- ❖ 快速排序： 就是将所有元素逐个转换为轴点的过程
- ❖ derangement： 任何元素都不在原位
比如，顺序序列循环移位
- ❖ 好消息： 不需要很多次交换，即可使任一元素转换为轴点
- ❖ 问题： 如何交换？成本多高？

