

排序

快速排序：轴点

14-A1

邓俊辉

deng@tsinghua.edu.cn

左朱雀之茕茕兮，右苍龙之躍躍

分而治之

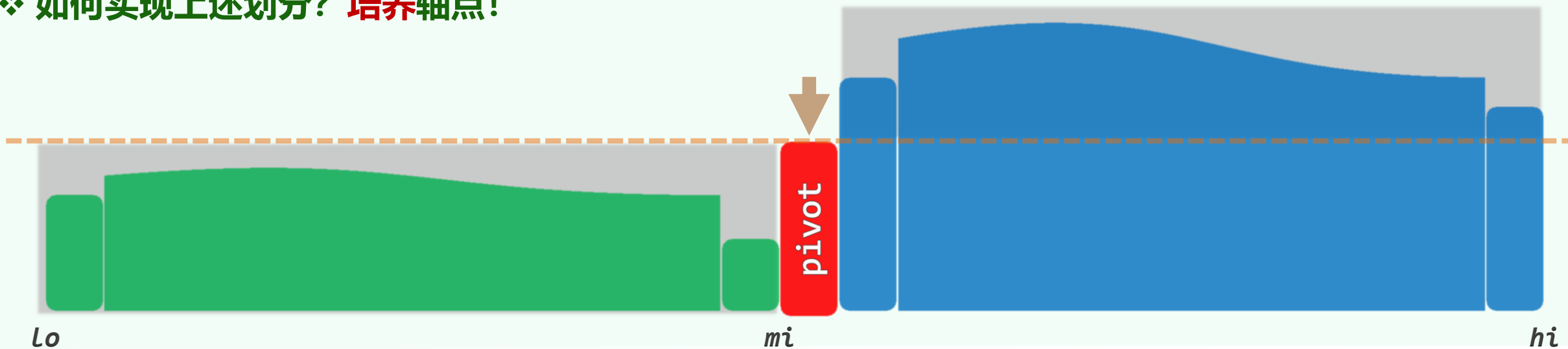
- ❖ **pivot**: $\max[lo, mi) \leq [mi] \leq \min(mi, hi)$
- ❖ 前缀、后缀各自（递归）排序之后，原序列便自然有序
$$sorted(S) = sorted(S_L) + pivot + sorted(S_R)$$
- ❖ mergesort难点在于**合**，而quicksort在于**分**
- ❖ 如何实现上述划分？**培养轴点**！



C. A. R. Hoare

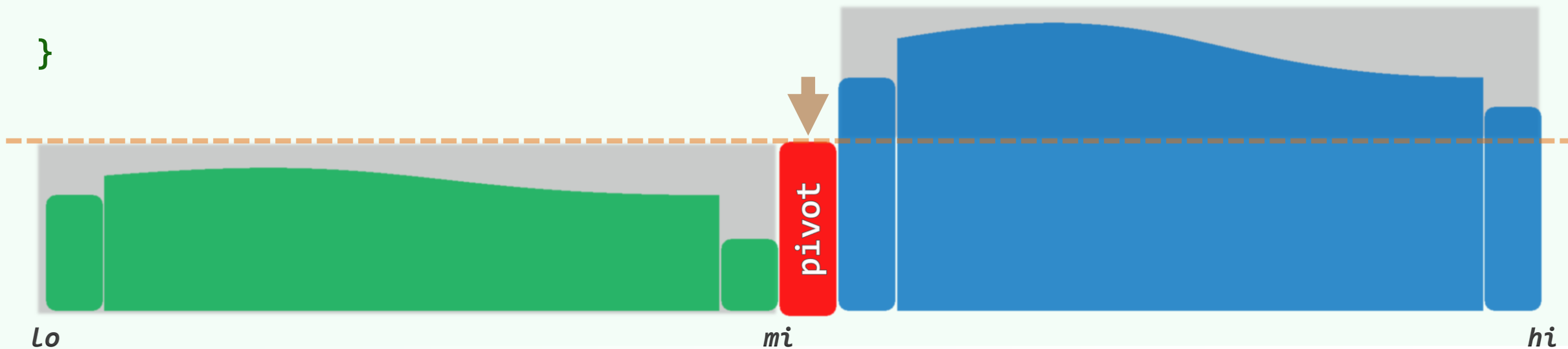
(1934 ~)

Turing Award, 1980



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

- ❖ 必要条件： 轴点必定已然**就位** // 尽管反之不然
- ❖ 进一步地： 一个序列有序，**当且仅当**所有元素皆为轴点
- ❖ 快速排序： 就是将所有元素**逐个转换**为轴点的过程
- ❖ 坏消息： 一个序列中**未必**总有轴点... 也就是所谓的
derangement： 任何元素都不在原位
比如，顺序序列循环移位
- ❖ 好消息： 不需很多**交换**，即可使**任一**元素转为轴点
- ❖ 问题： 如何交换？ 成本多高？

