

## 2. 向量

归并排序

复杂度

I think there is a world market  
for about five computers.

- T. J. Watson, 1943

邓俊辉

deng@tsinghua.edu.cn

❖ 算法的运行时间主要消耗于for循环，共有两个控制变量

初始： $j = 0, k = 0$

最终： $j = lb, k = lc$

亦即： $j + k = lb + lc = hi - lo = n$

❖ 观察：每经过一次迭代， $j$ 和 $k$ 中至少有一个会加一（ $j + k$ 也必至少加一）

❖ 故知：merge()总体迭代不过 $O(n)$ 次，累计只需线性时间！

❖ 这一结论与排序算法的 $\Omega(n \log n)$ 下界并不矛盾——毕竟这里的B和C均已各自有序

❖ 注意：待归并子序列不必等长

亦即：允许 $lb \neq lc, mi \neq (lo + hi)/2$

❖ 实际上，这一算法及结论也适用于另一类序列——列表（下一章）

## 综合评价

### ❖ 优点

实现最坏情况下最优 $O(n \log n)$ 性能的第一个排序算法

不需随机读写，完全顺序访问——尤其适用于

列表之类的序列

磁带之类的设备

只要实现恰当，可保证稳定——出现雷同元素时，左侧子向量优先

可扩展性极佳，十分适宜于外部排序——海量网页搜索结果的归并

易于并行化

### ❖ 缺点

非就地，需要对等规模的辅助空间——可否更加节省？

即便输入完全（或接近）有序，仍需 $\Theta(n \log n)$ 时间——改进...