02-F3

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

运行时间

❖ 二路归并中，两个while循环每迭代一步

i都会递增；j或k中之一也会随之递增

❖ 因：0 <= j <= lb，0 <= k <= lc

故：累计迭代步数 <= lb + lc = n

二路归并只需$\mathcal{O}(n)$时间

```
while ( ( j < lb ) && ( k < lc ) )
   A[i++] = ( B[j] <= C[k] ) ? B[j++] : C[k++];
while ( j < lb )
   A[i++] = B[j++];
```

（注意，即便lb和lc不相等，甚至相差悬殊，这一结论依然成立）

❖ 于是可知，归并排序的时间复杂度为$\mathcal{O}(n \log n)$

❖ 这一算法的思路及结论，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 $\mathcal{O}(n \log n)$ 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？
- 即便输入已是完全（或接近）有序，仍需 $\Omega(n \log n)$ 时间——如何改进？