

把两个有序的序列合并成一个的做法叫做**二路归并**

归并排序具体过程

由于递归版本比较难以理解,那我们就手造一下归并排序的过程

[5, 4, 9, 6, 6, 7, 4, 3]

归并排序的主要思想是:

依次排序区间长度为1, 2, 4...的相邻区间

那么第一次排序要**分别排序**的区间就是[1, 2], [3, 4], [5, 6], [7, 8]

排序后就是:[4, 5, 6, 9, 6, 7, 3, 4]

排完之后,长度为2的区间已经是有序的了,所以就可以再进行二路归并

这一次区间就是[1, 4], [5, 8]

那么排序完就是[4, 5, 6, 9, 3, 4, 6, 7]

前后分成了两个有序的序列

那么再进行排序,这一次就是整个区间了[1, 8]

再进行一次排序[3, 4, 4, 5, 6, 6, 7, 9]。

我们为什么要从最小区间进行二路归并

因为把小的区间排序成有序序列,才可以在大的区间进行二路归并

和递归的联系

假如我们定义 $merge(l, r)$ 为对区间 $[l, r]$ 中的数进行二路归并的话,

那么实际上, 归并排序就是在依次执行以下操作(可以对照上面的例子):

$merge(1, 2); merge(3, 4); merge(5, 6); merge(7, 8)$

$merge(1, 4), merge(5, 8)$

$merge(1, 8)$

这个 $merge$ 函数实际上就是咱们上课讲的那个二路归并函数

再仔细看一下我们的递归函数。

它实际上就是在模拟上述函数执行过程。

在最深层,执行 $merge(1, 2); merge(3, 4); merge(5, 6); merge(7, 8)$,

执行完上一层之后,开始执行 $merge(1, 4), merge(5, 8)$

所谓的“分解”,就是不断地往深层递归,并且确定要二路归并地边界

