

归并排序

将无序序列 s 进行排序。

本问题对无序序列 s 进行升序排序，排序后 s 是从小到大的。

在每一轮排序中，将 $s[0, \frac{n-1}{2}]$ 作为 **left**，将 $s[\frac{n-1}{2} + 1, n]$ 作为 **right**。然后对 **left** 和 **right** 分别进行排序。由于 **left** 和 **right** 是升序的，因此可以比较 **left** 和 **right** 中的第一个元素，将较小的一个放入新的数组 **t**，然后继续比较（假如 **left** 中第一个元素被取出，那么这时比较的就是 **left** 的第二个元素和 **right** 的第一个元素）。最终数组 **t** 即为升序的 **s**。

0	1	2	3	4	5	6	7	8	9
0	3	8	30	48	7	12	15	36	50
left					right				
t									

Diagram illustrating a binary tree structure. The root node is 0, which is shaded with blue diagonal lines. It has two children: node 3 (left) and node 8 (right). Node 3 has children 0 and 3. Node 8 has children 30 and 48. Node 0 has children 7 and 12. Node 3 has children 15 and 36. Node 48 has children 36 and 50. The tree is labeled 'left' and 'right' below the root node.

(2) $i = 1, j = 5, k = 1$ 时, $s[i] < s[j]$, 令 $t[k] = s[i]$, $i = i + 1 = 2, k = k + 1 = 2$;

0	1	2	3	4	5	6	7	8	9
0	3	8	30	48	7	12	15	36	50
left			right						
0	3								
t									

(3) $i = 2, j = 5, k = 2$ 时, $s[i] > s[j]$, 令 $t[k] = s[j]$, $j = j + 1 = 6, k = k + 1 = 3$;

0	1	2	3	4	5	6	7	8	9
0	3	8	30	48	7	12	15	36	50
left			right						
0	3	7							
t									

(4) 重复上面的步骤, 直到 s 中所有元素都被复制到新数组 t 中, 即 $i > 4$ 且 $j > 9$ 为止, 即可得到升序数组 s , 算法结束;

对于长度 n 的序列 s , 每一轮放置所需要的时间为 $O(n)$, 总共需要 $\log_2 n$ 轮, 该算法的时间复杂度为 $O(n \cdot \log_2 n)$ 。