

5 Merge Sort

归并排序

问题:

将无序序列 s 进行排序。

解法 1:

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。

对于长度为 n ，下标在左开右闭区间 $[0, n)$ 中的序列 s ，将其分为左（left）和右（right）两个部分，假设现在已经将 left 和 right 两个部分已经是升序的，那么只需要将 left 和 right 合并排序即可。同样的将 left、right 再次分为 left 和 right 两部分，用相同方式进行排序，这样递归下去，就可以得到升序序列 s 。

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** 进行合并排序，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**。

对于下面长度为 10 的无序序列 s 进行升序排序，我们只看最后一轮时的情况，即 left 和 right 已经是升序的情况，其中 s[0,4] 为 left，s[5,9] 为 right，设 i = 0，j = 5，k = 0 分别为 left、right 和排序后的新数组 t 的下标：

[illegible]

(1) $i = 0, j = 5, k = 0$ 时, 比较 left 和 right 的左边第一个元素, $s[i] < s[j]$, 令 $t[k] = s[i], i = i + 1 = 1, k = k + 1 = 1$;

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

(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 $O(n \cdot \log_2 n)$ 。