

第五章，分治算法

5.1.

非递归 合并算法

IterMergeSort(A, n)

1. $s \leftarrow 1$

2. while $s < n$ do

3. MergeAB(A, B, s, n) // 合并大小为s的相邻子数组，保存在B中

4. $S \leftarrow S + S$

5. MergeAB(B, A, s, n)

6. $S \leftarrow S + S$

7. MergeAB(A, B, s, n)

2.1. $j \leftarrow 1, t \leftarrow 1$

2. while $t \leq \frac{n}{2}$ do 每趟合并几次

3. $j \leftarrow j + 2S - 1$

4. merge(A, p, q) 对 p 到 q 进行归并

5. $t \leftarrow t + 1$

6. $j \leftarrow j + 2S$

5.3 利用二分法，将数组分开，舍去不在的范围

Search(A, p, r, x)

1. if $p > r$ then return 0

2. else

3. $q \leftarrow (p+r)/2$

4. if $A[q] = x$ then return q

5. else if $x < A[q]$ then

6. return Search(A, p, q-1, x)

7. else Search(A, q+1, r, x)

8. return

9. return

10. return

$$T(n) = \begin{cases} O(1) & \text{if } n=1 \\ T(n/2) + O(1) & \text{if } n \geq 2 \end{cases}$$

$$a=1, b=2 \rightarrow \log_2 a = 0$$

$$n^{\log_2 a} = n^0 = 1$$

$$\text{则 } T(n) = O(\log n)$$

智能科学与技术系

31520181154417

周雨

MergeSort(A, p, r)

if $p < r$ then

$q \leftarrow \lfloor (p+r)/2 \rfloor$

$i \leftarrow \lfloor (p+q)/2 \rfloor, j \leftarrow \lfloor (q+r)/2 \rfloor$

MergeSort2(A, p, i)

MergeSort2(A, i+1, q)

MergeSort2(A, q+1, j)

MergeSort2(A, j+1, r)

Merge2(A, p, i, j, q, r)

$$T(n) = 4T(n/4) + O(n)$$

$$T(n) = O(n \log n)$$

算法思想：

1. 选择一个元素 $a[q]$ 作为支点，将数组划分为 A_1, A_2, A_3 ，它们分别为包含

小于、等于和大于 $a[q]$ 的元素

2. 根据数组的长度可以判断第 k 小元素出现在哪个数组中的哪一个中，

对于包含第 k 小元素的那个数组递归调用算法进行处理，舍去不在的

两部分。

QuickSortMin(A, p, r, k)

if $p < r$ then

$q \leftarrow \text{Partition}(A, p, r)$

if $q = k$, then

return $A[q]$

else if $q < k$, then

QuickSortMin(A, q+1, r, k-q)

else QuickSortMin(A, p, q-1, k).

$$T(n) = \frac{1}{n} \sum_{k=1}^n (T(k) + T(n-k)) + n$$

$$= \frac{n+1}{n} T(n-1) + \frac{1}{n}$$

$$= \frac{n+1}{2} + \frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \dots + \frac{1}{2} + \frac{1}{1}$$

$$= O(n)$$