

分治法

讲授者 王爱 弱

目求 递归分析*i* 

公 分治法的思

人立山社

**决**读排序

惟和堆排序

大整数乘法 和矩阵乘法

#### 分治法

讲授者 王爱娟 aijuan321@foxmail.com

重庆理工大学 计算机科学与工程学院

August 22, 2024

## 目录

分治法

讲授者 王爱 娟

日東
湯归分析っ

分治法的 相

合并排序 快速排序

快速排序

大整数乘法 和矩阵乘法

- ① 递归分析方法
- 2 分治法的思想
- ③ 合并排序
- 4 快速排序
- 5 堆和堆排序
- 6 大整数乘法和矩阵乘法

### 递归的概念

分治法

讲授者 王爱 娟

日录

.归分析方 ·

分治法的思 想

合并排序

快速排序

大整数乘法

#### 引例

- ① 对于任意非负整数n, 计算阶乘函数F(n) = n!的值;
- ②  $\exists n > 1$  时,n! = 1 \* 2 \* ... \* (n-1) \* n = (n-1)! \* n;
- ③ 可以使用递归的方法计算F(n) = F(n-1) \* n。

### 递归的概念

分治法

讲授者 王爱 娟

目录

归分析方

分治法的思 想

<sup>②</sup> 合并排序

央速排序

大整数乘法 和矩阵乘法

#### 引例

- ① 对于任意非负整数n,计算阶乘函数F(n) = n!的值;
- ②  $\exists n > 1$  时,n! = 1 \* 2 \* ... \* (n-1) \* n = (n-1)! \* n;
- **③** 可以使用递归的方法计算F(n) = F(n-1) \* n。

#### 伪代码

- 1: function F(n)
- 2: if n = 0 then
- 3: return 1
- 4: else
- 5: **return** F(n-1) \* n
- 6: end if
- 7: end function

#### 乘法执行次数M(n)

- n > 0  $\forall$ , M(n) = M(n-1) + 1
- → 计算F(n-1)\*n需要1次乘法
- n = 0时, M(0) = 0, 不需要乘法
- 替换法: M(n) = M(n-i) + i = n

# 递归方程求解

分治法

讲授者 王爱 娟

目录

归分析ブ

分治法的思

-会并排序

'B 기 해기.

1/2 & 1/2 lik

大整数乘法 和矩阵乘法

#### 递归方程求解-替换法

例1: 计算
$$W(n)$$
,  $W(n) = W(n-1) + n - 1$ ,  $W(1) = 0$   
 $W(n) = W(n-1) + n - 1 = W(n-2) + (n-2) + (n-1)$   
 $= W(n-3) + (n-3) + (n-2) + (n-1) = \dots$   
 $= W(1) + 1 + 2 + \dots + (n-2) + (n-1) = 1 + 2 + \dots + (n-2) + (n-1)$   
 $= n(n-1)/2$ 

# 递归方程求解

分治法

讲授者 王爱 娟

目录

递归分析ス 法

分治法的思 想

合并排序 快速排序

堆和堆排序

#### 递归方程求解-替换法

例1: 计算
$$W(n)$$
,  $W(n) = W(n-1) + n - 1$ ,  $W(1) = 0$   
 $W(n) = W(n-1) + n - 1 = W(n-2) + (n-2) + (n-1)$   
 $= W(n-3) + (n-3) + (n-2) + (n-1) = \dots$   
 $= W(1) + 1 + 2 + \dots + (n-2) + (n-1) = 1 + 2 + \dots + (n-2) + (n-1)$   
 $= n(n-1)/2$ 

例2: 计算
$$W(n)$$
,  $W(n) = 2W(n/2) + n - 1$ ,  $n = 2^k$ ,  $W(1) = 0$ 

$$W(n) = 2W(2^{k-1}) + 2^k - 1 = 2[2W(2^{k-2}) + 2^{k-1} - 1] + 2^k - 1$$

$$= 2^2W(2^{k-2}) + 2^k - 2 + 2^k - 1$$

$$= 2^2[2W(2^{k-3}) + 2^{k-2} - 1] + 2^k - 2 + 2^k - 1$$

$$= 2^3W(2^{k-3}) + 2^k - 2^2 + 2^k - 2 + 2^k - 1 = \dots$$

$$= 2^kW(1) + k2^k - (2^{k-1} + 2^{k-2} + \dots + 2 + 1) = k2^k - 2^k + 1$$

$$= nlogn - n + 1$$

## 递归分析方法

分治法

讲授者 王爱 娟

目录

递归分析プ 法

分治法的思 想

合并排序

堆和堆排户

大整数乘法 和矩阵乘法

### 常见递归类型

#### 减一算法

- 算|法利用一个规模为n的实例和规模为n-1的给定实例之间的关系来对问题求解(插入排序,课本4.1)。
- T(n) = T(n-1) + f(n)

#### 减常因子算法

- 规模为n的实例化简为一个规模为n/b的给定实例来求解(俄式乘法)。
- $\bullet \ T(n) = T(n/b) + f(n)$

#### 分治算法

- 给定实例划分为若干个较小的实例,对每个实例递归求解,然后再把较小的实例合并成给定实例的一个解(快速排序,合并排序)。
- T(n) = aT(n/b) + f(n)



### 分治法的基本思想

分治法

讲授者 王爱 娟

目录 递归分析方

分治法的思 想

合并排序

厌逐排片

堆和堆排

大整数乘法 和矩阵乘法

#### 基本常识

小问题往往比大问题容易解决!



## 分治法的基本思想

讲授者 王愛

#### 基本常识

小问题往往比大问题容易解决!

#### 分治法的基本思想

- 将规模为N的问题分解为k个规模较小的子问题, 使这些子问题相互独立, 可分别求解
- ② 将k个子问题的解, 合并成原问题的解
- ◎ 如子问题的规模仍很大,则反复分解直到子问题小到可直接求解为止
- 子问题的解法通常与原问题相同, 导致递归过程

分治法的思 思

文 开排序 央速排序 崔和堆排序

# 分治法的基本思想

分治法

讲授者 王爱 娟

目录 递归分析: 法

分治法的足 想

合并排序 快速排序 收五比址

堆和堆排序 大整数乘法

#### 例子

- 计算n个数字 $a_1, a_2, ..., a_{n-1}, a_n$ 的和;
- 如果n > 1,可以把该问题划分为两个子问题,前一半的数字之和和后一半的数字之和;
- $a_1 + a_2 + \dots + a_{n-1} + a_n = (a_1 + \dots + a_{n/2-1}) + (a_{n/2} + a_{n/2+1} + \dots + a_n)$
- 类似地一直划分,直到含有一个元素,直接返回该元素 值:

# 分治法的时间效率分析

讲授者 王夢

### 主定理(具体证明详见P376—-3 分治法)

 $T(n) = a * T(\frac{n}{b}) + f(n) \not = T(1) = c, n = b^k, a \ge 1, b \ge 2, c > 0,$ 如果 $f(n) \in \Theta(n^d), d > 0$ ,

- $\bullet$  if a < b<sup>d</sup>, T(n)  $\in \Theta(n^d)$
- $\bullet$  if  $a = b^d$ ,  $T(n) \in \Theta(n^d \log n)$
- $\bullet$  if  $a > b^d$ ,  $T(n) \in \Theta(n^{\log_b^a})$

注释:

- $T(n) = aT(n/b) = a^2T(n/b^2) = a^{\log_b n}T(1) = cn^{\log_b a}$ (课本 附录A定理)
- ② 本质上是比较aT(n/b)和f(n)增长速度,并利用P43定理;



# 分治法的时间效率分析

分治法

讲授者 王爱 娟

目录

归分析方

分治法的思 <sup>相</sup>

-合并排序

|大型和F月

大整数乘法 和矩阵乘法

### 主定理应用例子

求解如下递归方程

$$T(n) = 9T(n/3) + n$$

$$T(n) = T(n/2) + n$$

$$T(n) = T(n/3) + 1$$

# 分治法的时间效率分析

### 主定理应用例子

求解如下递归方程

- T(n) = 9T(n/3) + n
- T(n) = T(n/2) + n
- **3** T(n) = T(n/3) + 1

### 答案

- ①  $a = 9, b = 3, f(n) = n \in \Theta(n)$ , 即d = 1, 由于 $a = 9 > b^d = 3^1$ , 所以 $T(n) \in \Theta(n^{log_b^a}) = \Theta(n^2)$
- ③  $a=1,b=3,f(n)=1\in\Theta(1)$ ,即d=0,由于 $a=1=b^d=3^0$ ,所以 $T(n)\in\Theta(n^dlogn)=\Theta(logn)$

讲授者 王爱 娟

> b 分治法的思

合并排序 快速排序 惟和堆排序 大整数乘法



## 合并排序

讲授者 王爱 娟

目录

归分析方

分治法的思 想

合并排序

快速排序

大整数乘法

#### 问题

将n个元素排成非递减顺序

# 合并排序

分治法

讲授者 王爱 娟

> 造归分析方 失 ↑治法的思

并排序

快速排序 堆和堆排

#### 问题

将n个元素排成非递减顺序

#### 解题思路

- 若n为1,算法终止
- ② 将n个待排元素分割成K个大致相等的子序列。这里 令K=2,两个子序列分别为B和C
- 分别对它们排序
- 然后将B与C合并为一个有序序列

分治法

讲授者 王爱 娟

目录

日分析方

分治法的思 <sup>相</sup>

合并排序

快速排户

15 6 15 1

大整数乘法 和矩阵乘法 例子: 合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 



分治法

讲授者 王爱 娟

目录 递归分析フ

分治法的 思

合并排序

快速排序

大整数乘法 和矩阵乘法

#### 例子:合并两个有序集B和C

 $B=\{1,2,6,9\}, C=\{4,5,7,8,13,16\}$ 

#### 合并过程

将C中的元素依次插入到B中的相应位置即可



讲授者 王爱

目录 递归分析方 去

分冶法的心想 合**并排序** 快速排序

堆和堆排序 大整数乘法 和矩阵乘法 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

B 1 2 6 9

4 5 7 8 13 16



分治法 讲授者 王爱

目录 選归分析方 よ

企 分治法的思 想

**合并排序** 快速排序 堆和堆排序

大整数乘法和矩阵乘法

例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

В

1 2 4 6 9

(

5 | 7 | 8 | 13 | 16



讲授者 王爱

目录 単归分析方 よ

分治法的思 想 合并排序

◆ 开排序 快速排序 作和堆排序 七 教教乘法 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

В

1 2 4 5 6 9

(

7 8 13 16



方治法 讲授者 王爱

1录 造归分析方 失

分治法的思 想 合并排序

央速排序 作和堆排序 1-数数乘注 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

В

1 2 4 5 6 7 9

(

8 | 13 | 16



分治法 讲授者 王爱

1录 造归分析方 去

想 合并排序 快速排序

位和堆排序 大整数乘法 和矩阵乘法 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

В

1 2 4 5 6 7 8 9



讲授者 王爱

1录 造归分析方 去

想 合并排序 快速排序

体 堆和堆排序 大整数乘法 和矩阵乘法 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

В

1 2 4 5 6 7 8 9 13

\_رَ



讲授者 王爱

1录 追归分析方 失

分治法的思 想 合并排序

快速排序 作和堆排序

大整数乘法 和矩阵乘法 例子:合并两个有序集B和C

 $B = \{1,2,6,9\}, C = \{4,5,7,8,13,16\}$ 

合并过程

将C中的元素依次插入到B中的相应位置即可

8

9

В

1 2 4 5 6



end while

end while

end while

while i < len(B) do

while i < len(C) do

A[k] = B[i];k++;i++;

A[k] = C[i]; k++; i++;

```
分治法
讲授者 王爱
```

```
merge(B,C,A)有序集合合并伪代码 i=0;\ j=0;\ k=0 while i<len(B) and j<len(C) do if B[i]\leq C[j] then A[k]=B[i];\ i++ else A[k]=C[j];\ j++ end if k=k+1
```



## merge的时间效率分析

万治法

讲授者 王爱 娟

日求 递归分析:

分治法的思

合并排序

央速排序

大整数乘法 和矩阵乘法 如果两个集合B和C都是有序的,那么

- ① merge算法返回的A也是有序的
- ② 算法的基本操作是"比较"
- 3 算法只需要遍历B一次、依次插入C的元素即可
- 4 算法时间开销是Θ(n),效率高



## merge的时间效率分析

分治法

讲授者 王爱 娟

目录 递归分析方 法

分治法的思 想

↑开排序 快速排序 ŧ和堆排序

大整数乘法和矩阵乘法

如果两个集合B和C都是有序的,那么

- ① merge算法返回的A也是有序的
- ② 算法的基本操作是"比较"
- ③ 算法只需要遍历B一次,依次插入C的元素即可
- 算法时间开销是Θ(n),效率高

#### 合并排序的关键问题

- 规模为n的数组A划分为大小相等的两个集合B和C
- 如果B和C有序,则直接使用merge即可完成对A的排序
- 如何实现B和C有序



## merge的时间效率分析

万治法

讲授者 王爱 娟

目录 递归分析方 法

分治法的思 想

→并排序 快速排序 ←和推排

大整数乘; 和矩阵乘; 如果两个集合B和C都是有序的,那么

- ① merge算法返回的A也是有序的
- ② 算法的基本操作是"比较"
- 3 算法只需要遍历B一次,依次插入C的元素即可
- 算法时间开销是Θ(n), 效率高

#### 合并排序的关键问题

- 规模为n的数组A划分为大小相等的两个集合B和C
- 如果B和C有序,则直接使用merge即可完成对A的排序
- 如何实现B和C有序



## 递归地对B和C进行排序策略

分治法

讲授者 王爱 娟

目录 递归分析方

分治法的思 <sub>机</sub>

合并排序

快速排月

大整数乘法

#### 常识

当数组长度为1时,数组天然有序

## 递归地对B和C进行排序策略

讲授者 王爱

#### 常识

当数组长度为1时,数组天然有序

#### 对B和C进行排序策略

- 对B和C进行不断的划分,产生更小的子集,直到子集的 大小为1
- ② 大小为1的子集天然有序,对它们采用merge方法合并, 产生更大的有序集合
- ◎ 依次对新产生的有序集合采用merge方法合并,最终可完成对B和C的排序,从而保证它们有序

分治法

井授者 王愛

日忌

归分析方

分治法的思 <sub>相</sub>

合并排序

快速排序

IX ACTIFY.

大整数乘法和知晓乘法

#### 例子

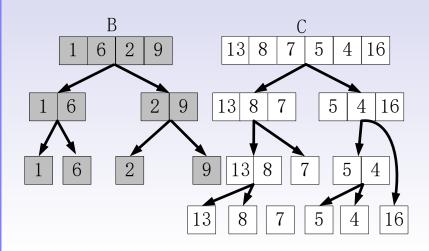
$$B = \{1,6,2,9\} \ C = \{13,8,7,5,4,16\}$$



讲授者 王爱

例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 

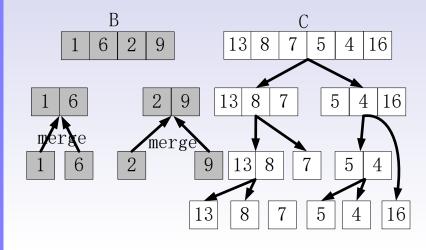




讲授者 王爱

例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 





分治法

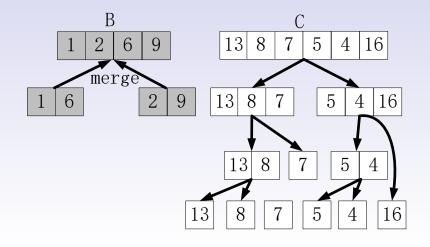
讲授者 王爱 娟

4 小 递归分析方 去

合并排序

大还排户 作和堆排序 大整数乘法 例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 





分治法 讲授者 王爱

> **娟** 1录

法分治法的思

**分并排序** 央速排序

· 快速排序 全和堆排序

理和理排序 大整数乘法 和矩阵乘法 例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 

1 2 6 9

13 8 7 5 4 16

7 | 8 | 13

4 | 5 | 16

0 13

7

 $\begin{bmatrix} 1 & 1 & 2 \\ 5 & 4 \end{bmatrix}$ 



分治法

讲授者 王愛

目录 递归分析方

.. 分治法的思

**今并排序** 

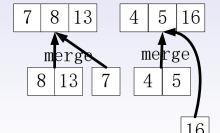
央速排序 作和堆排序

大整数乘法 和矩阵乘法 例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 

1 2 6 9

13 8 7 5 4 16



分治法

讲授者 王爱

目录 递归分析方

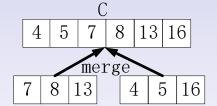
分治法的思 <sub>相</sub>

合并排序 血油排皮

人~~~// 准和堆排序

大整数乘法 和矩阵乘法 例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 





## 例子

分省法

讲授者 王爱 娟

目录

へ 分治法的ほ

分治法的思 想

合并排序

快速排序

堆和堆排

大整数乘法和矩阵乘法

例子

 $B = \{1,6,2,9\} C = \{13,8,7,5,4,16\}$ 

В

1 2 6 9

C

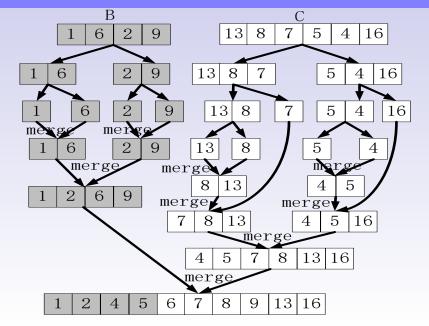
4 5 7 8 13 16



### 归并排序过程总览



方思



### 合并排序的递归算法

```
讲授者 王爱
```

目录 递归分析方 法

合并排序

```
合并排序算法MergeSort(A[0..n-1])
```

```
//輸入: 未排序序列A[0..n-1]
//輸出: 己排序序列A[0..n-1]

if n > 1 then
    copy A[0, \lfloor \frac{n}{2} \rfloor - 1] to B[0, \lfloor \frac{n}{2} \rfloor - 1]
    copy A[\lfloor \frac{n}{2} \rfloor, n - 1] to C[0, \lceil \frac{n}{2} \rceil - 1]
    MergeSort(B[0, \lfloor \frac{n}{2} \rfloor - 1])
    MergeSort(C[0, \lceil \frac{n}{2} \rceil - 1])
    merge(B,C,A)
end if
```

▶ Skip merge alogirthm



### MergeSort算法解读

分治法

讲授者 王爱 娟

日求

归分析方

分治法的思 想

合并排序 决速排序

堆和堆排序 大整数乘法 ● 数组A被对半划分为子数组B和C

② B和C被递归地划分为更小的子集,直到子集大小为1

● merge方法被调用合并有序的子集,直至所有集合被合并

● 数组A排序完成

overview

## MergeSort算法的时效分析

分治法

讲授者 王爱 娟

> ・か 5归分析ス <del>-</del>

分治法的思 想

央速排序 佳和堆排序

大整数乘法和矩阵乘法

- ❶ 算法的基本操作是"比较"
- ② 设 $n=2^k$ ,则关键字比较次数的递推关系式为

$$C(n) = 2C(\frac{n}{2}) + C_{merge}(n), \quad C(1) = 0$$

**③** 在最坏情况下,merge的效率 $C_{merge}(n) = n-1$ ,

$$C(n) = 2C(\frac{n}{2}) + n - 1, \quad C(1) = 0$$

$$\Rightarrow C(n) \in \Theta(n \log n)$$

▶ theorem

当 $n=2^k$ 时,可以求得最差效率C(n)递推式的精确解: $C(n)=nlog_2n-n+1$ 



分治法

讲授者 王愛 娟

目录

分治法的思 想

快速排序 堆和堆排戽

堆和堆排序大整数乘法

#### 问题

- 问题规模的分解:将规模为n的数组 $A \Rightarrow$  两个规模为 $\frac{n}{2}$ 的子问题Bac
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需对它们进行排序,它们本身就是所需的解
- 利用已有的解,求取更大问题的解:基于子集,采用一些操作(merge),产生更大的有序集合
- ▼取原问题的解: 反复操作,直至所有子集被合并完成, 即求得原问题的解!



分治法

讲授者 王夢

#### 问题

- 问题规模的分解:将规模为n的数组A ⇒ 两个规模为 $\frac{n}{2}$ 的 子问题B和C
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需 对它们进行排序,它们本身就是所需的解
- 利用已有的解, 求取更大问题的解: 基于子集, 采用一些 操作(merge),产生更大的有序集合
- 求取原问题的解: 反复操作, 直至所有子集被合并完成, 即求得原问题的解!



# 分治法 进授者 王爱

#### 问题

- 问题规模的分解:将规模为n的数组 $A \Rightarrow$  两个规模为 $\frac{n}{2}$ 的子问题Bac
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需对它们进行排序,它们本身就是所需的解
- 利用已有的解, 求取更大问题的解: 基于子集, 采用一些操作(merge), 产生更大的有序集合
- 求取原问题的解: 反复操作, 直至所有子集被合并完成, 即求得原问题的解!



# 分治法 进授者 王爱

#### 问题

- 问题规模的分解:将规模为n的数组 $A \Rightarrow$  两个规模为 $\frac{n}{2}$ 的子问题Bac
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需对它们进行排序,它们本身就是所需的解
- 利用已有的解, 求取更大问题的解: 基于子集, 采用一些操作(merge), 产生更大的有序集合
- 求取原问题的解: 反复操作, 直至所有子集被合并完成, 即求得原问题的解!



分治法

讲授者 王夢

### 分治法之归并排序总结

#### 问题

- 问题规模的分解:将规模为n的数组 $A \Rightarrow$  两个规模为 $\frac{n}{2}$ 的子问题Bac
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需对它们进行排序,它们本身就是所需的解
- 利用已有的解, 求取更大问题的解: 基于子集, 采用一些操作(merge), 产生更大的有序集合
- 求取原问题的解:反复操作,直至所有子集被合并完成, 即求得原问题的解!



分治法

讲授者 王夢

### 分治法之归并排序总结

#### 问题

- 问题规模的分解:将规模为n的数组 $A \Rightarrow$  两个规模为 $\frac{n}{2}$ 的子问题Bac
- 继续分解: B和C的规模仍较大, 继续划分它们为更小的 子问题
- 停止分解: 当子问题的规模为1时, 停止划分
- 求子问题的解:由于规模为1的子问题,天然有序,无需对它们进行排序,它们本身就是所需的解
- 利用已有的解, 求取更大问题的解: 基于子集, 采用一些操作(merge), 产生更大的有序集合
- 求取原问题的解:反复操作,直至所有子集被合并完成, 即求得原问题的解!



## 快速排序

分治法 讲授者 王爱 娟

目录 単归分析方 Ŀ

分治法的思 想 合并排序

快速排序

大整数乘法和矩阵乘法

#### 回顾:中值问题

求一个数组(大小为n)的第k个最小元素。当 $k=\lceil n/2 \rceil$ 时,该元素被称为中值。

#### 解决办法:划分

- 将数组划分为两部分:一部分比某一值大,一部分比某一值小。
- $\mbox{th}/\mbox{1}, \{4,1,10,9,7,12,8,2,15\} \Rightarrow \{1,2\}, \{4\}, \{9,7,12,8,10,15\}$
- 可见,通过划分发现4是数组A中的第三小元素,并且它 正位于数组A中的第二位置,即A[2]=4。
- 这表明: 一次划分可将一个数字放在(有序)数组A中的正确位置。

### 快速排序

讲授者 王夢

回顾:中值问题

求一个数组(大小为n)的第k个最小元素。当k=[n/2]时,该元 素被称为中值。

### 解决办法:划分

- 将数组划分为两部分: 一部分比某一值大, 一部分比某一 值小。
- $\not\vdash t \not\vdash m$ ,  $\{4,1,10,9,7,12,8,2,15\} \Rightarrow \{1,2\}, \{4\}, \{9,7,12,8,10,15\}$
- 可见,通过划分发现4是数组A中的第三小元素,并且它 正位于数组A中的第二位置,即A[2]=4。
- 这表明: 一次划分可将一个数字放在(有序)数组A中的正 确位置。

### 应用划分于排序

反复划分,就可确定所有元素在数组A中的正确位置



分治法

讲授者 王爱 娟

EI 쿤

归分析方

分治法的思

合并排序

快速排序

**住和堆排**/

大整数乘法 和矩阵乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。
- ❸ 终止条件:被划分的数组长度为1,即A[0···s-1]的长度为1,A[s+1···n-1]长度为1.



分治法

讲授者 王愛 娟

目录

归分析方

分治法的思

想

合并排序

快速排序

作和堆排月

大整数乘法 和矩阵乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。
- ❸ 终止条件:被划分的数组长度为1,即A[0···s-1]的长度为1,A[s+1···n-1]长度为1.



分治法

讲授者 王爱 娟

目录

归分析方

分治法的思

11 (4) 111 (4)

[和堆排]

大整数乘法 和矩阵乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。
- ❸ 终止条件:被划分的数组长度为1,即A[0···s-1]的长度为1,A[s+1···n-1]长度为1.



分治法

讲授者 王爱 娟

目录

归分析方

分治法的是

合并排序

快速排序

和堆排月

大整数乘法 和矩阵乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。
- ❸ 终止条件:被划分的数组长度为1,即A[0···s-1]的长度为1,A[s+1···n-1]长度为1.



分治法

讲授者 王爱 娟

目录

法

分治法的思想 想

合并排序

主和堆排序

大整数乘法 和矩阵乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。
- ③ 终止条件:被划分的数组长度为1,即A[0···s-1]的长度为1,A[s+1···n-1]长度为1.

分治法

讲授者 王爱 娟

目录

法 分治法的思 <sub>想</sub>

合并排序 央速排序

堆和堆排序 大整数乘法

#### 算法思路

- 分区:取A中的一个元素P为支点(pivot),将A[0···n-1]划 分成3段: A[0···s-1],P和A[s+1···n]。
  - P(=A[s]), 它在有序序列A的s位置(一个正确位置)
  - A[0···s-1], 它中任一元素<P
  - A[s+1···n-1], 它中的任一元素>P
- ② 递归求解:分别递归地对A[0···s-1]和A[s+1···n-1]进行划分。

## 例子

分治法

讲授者 王爱 娟

日号

归分析方

分治法的思

人五排皮

快速排序

MEMP)

大整数乘法

采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序



采用划分的方式对数组A={4,1,10,9,7,12,8,2,15}进行排序



















采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序

































采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序







































采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序



















15

15

























分治法

讲授者 王聚 娟

目录 递归分析>

分治法的思

合并排序

10.00

堆和堆排

采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序



















(2)

















15

9

















分治法

讲授者 王爱 娟

目录 递归分析:

分治法的思

合并排序

伏还併厅

堆和堆排)

大整数乘法 和矩阵乘法 采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序



















(2)



































## 例子

分治法

讲授者 王爱 娟

目录 递归分析 フ

分治法的思

快速排序

堆和堆排

大整数乘法 和矩阵乘法 采用划分的方式对数组 $A=\{4,1,10,9,7,12,8,2,15\}$ 进行排序

4 1 (1

10

9 7

(12)

(8

 $\overbrace{15}$ 

15

10

10

9

 $\left(7\right)$ 

8

12

(15)

9)

10

(

(12) (

15

12) (

15

可以看出, 每一次划分都可确定一个元素的位置

### 快速排序算法

```
分治法
```

```
讲授者 王爱
娟
```

1录 <sup>進归分析方</sup>

法 分治法的思 想

合并排斥

**住和堆排**序

堆和堆排户 大整数乘?

```
快速排序算法QuickSort(A[l···r])
```

```
//使用快速排序法对序列或者子序列排序 
 //输入: 子序列A[l..r]或者序列本身A[0..n-1] 
 //输出: 非递减序列A 
 if l < r then 
 s = partition(A[l \cdots r]) 
 QuickSort(A[l \cdots s-1]) 
 QuickSort(A[s+1 \cdots r]) 
 end if
```

### 快速排序算法

```
讲授者 王爱
```

```
快速排序算法QuickSort(A[l···r])
```

```
//使用快速排序法对序列或者子序列排序
//输入: 子序列A[l..r]或者序列本身A[0..n-1]
//输出:非递减序列A
  if 1 < r then
     s = partition(A[l \cdot \cdot \cdot r])
     QuickSort(A[l \cdot \cdot \cdot s-1])
     QuickSort(A[s+1\cdots r])
  end if
```

### partition: 快速排序算法的核心

上述代码表明, 快速排序算法的核心是partition, 即如何对数 组A进行划分!

分治法

讲授者 王爱 绢

1 录

归分析方

分治法的思 想

合并排序

快速排序

堆和堆排片

大整数乘法和矩阵乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比 较
- 若当前元素小于P, 则置于数组前段, 否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
  - s的位置特点: A[s] < P, A[s+1] ≥ P</li>
  - 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置

分治法

讲授者 王愛 娟

1录

归分析方

分治法的思 想

合并排序

伏还併厅

堆和堆排户

大整数乘法和拓阵乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比 较
- 若当前元素小于P,则置于数组前段,否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
  - s的位置特点:  $A[s] < P, A[s+1] \ge P$
- 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置

$$\begin{array}{c|cccc} I & & & r \\ \hline P & \langle P & \rangle = P & ? & \end{array}$$

分治法 讲授者 王愛

目录

归分析方

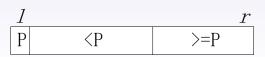
分治法的思 想

合并排序

央選排序

大整数乘法和矩阵乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比 较
- 若当前元素小于P,则置于数组前段,否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
  - s的位置特点:  $A[s] < P, A[s+1] \ge P$
  - 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置



分治法

讲授者 王爱 娟

1录

归分析方

分治法的思 想

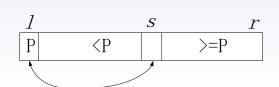
合并排序

快速排序

惟和堆排序

大整数乘法 和知時乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比较
  - 若当前元素小于P,则置于数组前段,否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
- s的位置特点: A[s] < P, A[s+1] ≥ P</li>
- 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置



分治法 讲授者 王爱

1录

分治法的思

合并排序 快速排序

大整数乘法 和矩阵乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比 较
- 若当前元素小于P,则置于数组前段,否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
  - s的位置特点: A[s] < P, A[s+1] ≥ P</li>
  - 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置

$$\begin{array}{c|cccc}
I & S & F \\
\hline
A[s] & \langle P & P & \rangle = P
\end{array}$$



分治法 讲授者 王爱

1录

云 分治法的思 想

合并排序 快速排序 准和堆排序

大整数乘法 和矩阵乘法

- 从左至右地扫描数组,在扫描过程中,将每个元素与P比较
  - 若当前元素小于P,则置于数组前段,否则置于后端
- 扫描完后,数组A划分为两个连续区域:小于P的区域和 大于等于P的区域
- s的位置特点: A[s] < P, A[s+1] ≥ P</li>
- 交换A[s]与A[l], 即完成划分
- A[s]正位于一个有序的数组A的正确位置



分治法

讲授者 王爱

目录

运 分治法的思

想

合并排序

快速排序

难和难排片

大整数乘法 和矩阵乘法

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
  - 若指针不相交,交换A[i]和A[j]。
  - 然后继续各自的扫描。



分治法

讲授者 王爱 娟

目录

分治法的思 想

合并排序

快速排序

堆和堆排)

大整数乘法 和矩阵乘法

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
  - 若指针不相交,交换A[i]和A[j]。
  - 然后继续各自的扫描。



分治法

讲授者 王爱 娟

1录

分治法的思 <sub>相</sub>

合并排序

快速排序

作和堆排)

大整数乘法 和矩阵乘法

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
- 若指针不相交,交换A[i]和A[j]。
- 然后继续各自的扫描。



分治法

讲授者 王愛 娟

目录

分治法的思 想

合并排序

人,还作月。

上数数面。

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
- 若指针不相交,交换A[i]和A[j]。
- 然后继续各自的扫描。



分治法

讲授者 王愛 娟

目录

分治法的思 想

合并排序

人,还作月。

上数数面。

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
- 若指针不相交,交换A[i]和A[j]。
- 然后继续各自的扫描。



分治法

讲授者 王爱 娟

录归分析方

分治法的思 思

央速排序

大整数乘法 和矩阵乘法

#### 快速排序的划分: 双向扫描

- 指针j从数组右边开始扫描,忽略大于中轴的元素,遇到 小于等于中轴的元素A[j]时停止
- 指针i从数组左边开始扫描,忽略小于中轴的元素,遇到 大于等于中轴的元素A[i]时停止
  - 若指针不相交,交换A[i]和A[j]。
  - 然后继续各自的扫描。

#### 两种扫描方式的比较

- Lomuto划分是单向扫描: 从左至右扫描数组
- 快速排序使用的划分是双向扫描,但先从右端开始扫描, 然后在从左端开始扫描,效率更高



分治法

讲授者 王爱 娟

目录

ョ分析方

分治法的思 想

合并排序

快速排月

堆和堆排

大整数乘法 和矩阵乘法

#### 例子



介治法

讲授者 王爱 娟

1录

日分析方

分治法的是 <sub>想</sub>

合并排序

IN TEMPT

大整数乘法

#### 例子





治治法

讲授者 王爱 娟

目录

日分析方

法

分治法的。 想

并排序

伏还作厅

堆和堆排。

大整数乘》 和矩阵乘》

#### 例子





介治法

讲授者 王爱 娟

目录

日分析方

分治法的是 <sub>相</sub>

合并排序

快逐排片

堆和堆排.

大整数乘》 和矩阵乘》

#### 例子





分治法

讲授者 王爱 娟

目录

日分析方

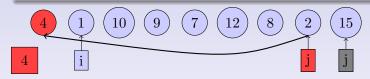
分治法的是

人名托拉克

6 开 4 万

大整数乘法

#### 例子





治法

讲授者 王爱

1 录

日分析方

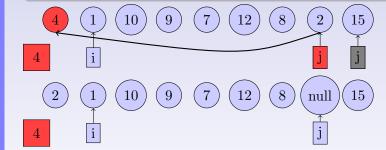
分治法的思 <sup>細</sup>

**}**并排序

快速排序

大整数乘法 和矩阵乘法



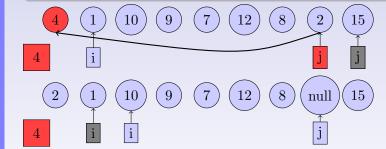




讲授者 王爱

例子







治法

讲授者 王爱 绢

目录

日分析方

公公法的田

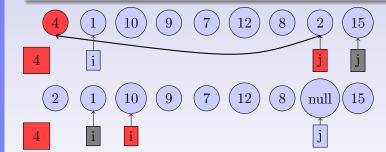
(J)

央速排序

大还#P 作和堆排)

大整数乘法 和矩阵乘法







治法

讲授者 王爱 錩

目录

日分析方

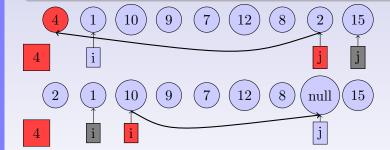
分治法的思

合并排序

快速排序

堆和堆排/ 大整数乘/







治法

讲授者 王爱 娟

录

ョ分析方

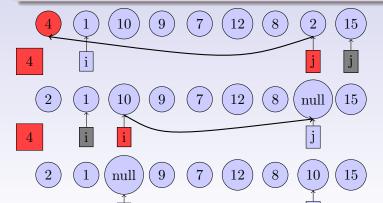
分治法的思 <sub>租</sub>

**}**并排序

快速排序

大整数乘法 和矩阵乘法







分治法

讲授者 王爱

目录

归分析方

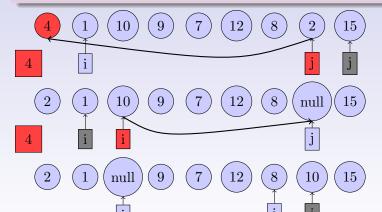
分治法的思 想

**今并排序** 

快速排序

大整数乘法 和矩阵乘法







治法

讲授者 王爱 娟

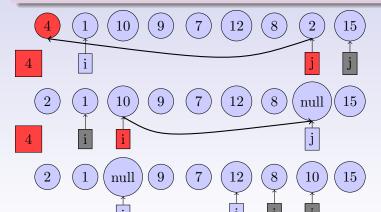
目录

思 合并排序

央速排序

世和世孫 大整数乘法 和矩阵乘法

#### 例子





分治法

讲授者 王爱

目录

日分析方

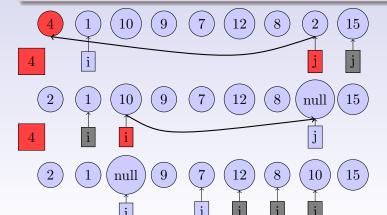
。 分治法的思

3. 合并排序

央速排序

堆和堆排/ 大整数乘?







分治法

讲授者 王爱

目录

自分析方

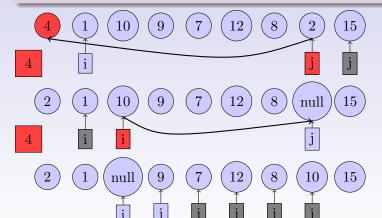
分治法的思 <sub>相</sub>

· 并排序

快速排序

大整数乘法 和矩阵乘法







治法

讲授者 王愛 娟

目录

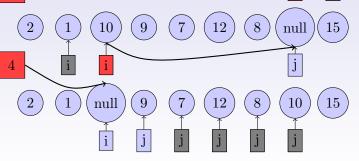
自分析方

分治法的思

合并排序

快速排序

大整数乘法 和矩阵乘法 例子 采用双向扫描的方式对数组A={4,1,10,9,7,12,8,2,15}进行划分 4 1 10 9 7 12 8 2 15 j j





分治法

讲授者 王爱 娟

目录

日分析え

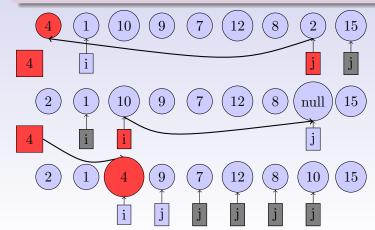
分治法的思

· 今并排序

央速排序

堆和堆排/ 大整数乘?





# 快速排序的划分算法

```
分治法
讲损者 王灵
```

```
讲授者 王爱
娟
```

```
法
分治法的思
想
```

```
合并排序
快速排序
除和操排序
```

```
堆和堆排序
大整数乘法
```

```
partition算法
```

```
//输入:数组A[low,···,high]
//输出:划分数组A
  i = low; j = high; x = A[low]
  while i < j do
     while i < j and A[j] > x do
       i = i - 1;
     end while
    A[i] = A[i]; i=i+1
     while i < j and A[i] < x do
       i = i + 1:
     end while
    A[j] = A[i]; j=j-1
  end while
  A[i] = x;
```

# 快速排序效率分析

分治法 讲授者 王愛

录

造归分析方 k

合并排序

堆和堆排序 大整数乘法 和矩阵乘法

#### 最坏情况下的效率分析

- 最差情况:所有分裂点都趋于极端:两个子数组中,有一个为空,另一个子数组仅仅比被划分数组少了一个元素
  - 基本操作: 比较
- 考虑一个严格递增的数组
- 进行了n+1次比较后,数组A[0,···,n-1]被划分为P(A[0])和A[1,···,n-1]
- 进 行 了n次 比 较 后 , 数 组A[1,···,n-1]被 划 分 为A[1]和A[2,···,n-1]
- 如此反复,直到最后一个子数组A[n-2,n-1],进行3次比较 后结束
- 总的比较次数:

$$C_{worst}(n) = (n+1) + n + \dots + 3 = \frac{(n+1)(n+2)}{2} - 3 \in \Theta(n^2)$$

# 快速排序效率分析

分治法

讲授者 王爱 娟

目录

归分析フ

分治法的原 相

合并排序

央速排序 6 5 - 16 - 11 - 1

大整数乘法 和矩阵乘法

#### 最好情况下的效率分析

- 最优情况: 所有分裂点位于相应子数组的中点
- 基本操作: 比较
- 进行了n次比较后,数组 $A[0,\cdots,n-1]$ 被划分为P和两个大小近似相当的部分: $A[0,\cdots,s-1]$ 和 $A[s+1,\cdots,n-1]$
- 如此反复, 直至对数组的划分结束
- 总的比较次数:

$$C_{best}(n) = 2 * C_{best}(\frac{n}{2}) + n \in \Theta(n \log n)$$



# 快速排序效率分析

分治法

讲授者 王爱 娟

日录

归分析プ

分治法的》 <sub>机</sub>

合并排序

快速排序

堆和堆排序

大整数乘法

#### 平均情况下的效率分析

- 基本操作: 比较
- 总的比较次数:

$$C_{avg}(n) = \frac{1}{n} \sum_{s=0}^{n-1} [n+1 + C_{avg}(s) + C_{avg}(n-1-s)],$$

$$C_{avq}(0) = 0, C_{avq}(1) = 0,$$

$$\Rightarrow C_{avq}(n) \approx 2n \ln n \approx 1.39n \log_2 n \in \Theta(n \log n)$$



## 堆和堆排序

讲授者 王爱

堆

堆可以定义为一棵二叉树,并且满足下面的条件:

完备性(完全二叉树):除了最后一层最右边的元素可能缺, 其他都是满的:

堆特性: 每个节点的值大于等于它的子女节点的值, 且只存在

一棵n个节点的完全二叉树,它的高度等于 $|log_2n|$ ;

#### 堆操作与堆构造

插入操作:与自顶向下构造堆相似,最坏情况下的复杂度 为 $O(log_2n)$  (树的高度)

删除操作:要删除的节点和堆的最后一个节点交换,然后删 除:

插入、删除等操作复杂度与堆构造相同。

# 堆排序

分治法

讲授者 王爱 娟

目录

归分析》

分治法的思 <sub>相</sub>

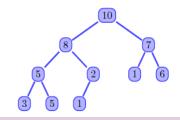
合并排序

堆和堆排序

大整数乘法 和矩阵乘法

#### 堆的特性和实现

- 只存在一棵n个节点的完全二叉树, 它的高度等于log2n;
- 可以用数据实现堆, 从上到下, 从左到右的方式记录堆的元素;
- 为了表达方便, 在数组的开始位置设置为空;
- 父母节点i位于数组的前[n/2]个位置,它的子女位于2i和2i+1;
- 对于一个位于 $i(2 \le i \le n)$ 的建来说,它的父母将会位于[i/2]。



堆的数组表示

(	0 :		1 2		3	4	5
		1	.0	8	7	5	2
	6		7 8		9	1	0
		1	6	3	5	1	L



## 堆的构造

分治法

讲授者 王爱 娟

目录 递归分析方 法

分治法的思 想

合并排序 中读排序

堆和堆排)

大整数乘法 和矩阵乘法 堆的构造-自底向上:

方法要点

- 按照给定的顺序放置节点;
- ② 从最后的父母节点K开始,到根为止,检查节点是否满足要求;如果不满足,交换K和子女最大键值节点的位置;
- ◎ 检查新的位置上,执行步骤2,检查键值K是不是满足要求;
- 重复步骤2和3, 直到对树的根完成操作。

实例: 对列表2,9,7,6,5,8构造堆



## 堆的构造

分治法

讲授者 王爱 娟

目录 递归分析方 法

分治法的思 想

合开排序

堆和堆排序

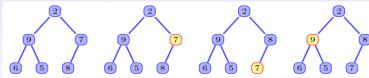
大整数乘法 和矩阵乘法

#### 堆的构造-自底向上:

#### 方法要点

- 按照给定的顺序放置节点;
- ② 从最后的父母节点K开始,到根为止,检查节点是否满足要求;如果不满足,交换K和子女最大键值节点的位置;
- ◎ 检查新的位置上,执行步骤2,检查键值K是不是满足要求;
- 重复步骤2和3, 直到对树的根完成操作。

#### 实例: 对列表2,9,7,6,5,8构造堆





## 堆的构造

分治法

讲授者 王爱 娟

目录 递归分析方 4

分治法的思 <sub>想</sub>

合并排序

生和堆排序

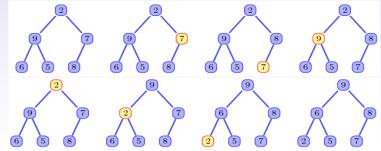
大整数乘法 和矩阵乘法

#### 堆的构造-自底向上:

方法要点

- 按照给定的顺序放置节点;
- ❷ 从最后的父母节点K开始,到根为止,检查节点是否满足要求;如果不满足,交换K和子女最大键值节点的位置;
- ❸ 检查新的位置上,执行步骤2,检查键值K是不是满足要求;
- 重复步骤2和3, 直到对树的根完成操作。

#### 实例: 对列表2,9,7,6,5,8构造堆



# 堆排序

分治法

讲授者 王爱 娟

日 ※ 遊归分析方法 分治法的思

堆和堆排序 大整数乘法

#### 堆排序算法

第一步: 构造堆;

第二步: 删除最大键值, 即对剩下的堆应用n-1次根删除操作。

重复一二步, 最终结果就是按照降序删除了的数据元素。

#### 时间效率分析

堆构造阶段的时间效率属于O(n);

第二阶段就是把堆的规模从n消减到2的过程,消根所需要的键值比较次数为 $C(n) < 2nlog_2n$ ;

堆排序整个过程的时间效率为O(nlogn)。



分治法

讲授者 王爱 娟

日忌

5归分析方 -

分治法的思 想

合并排序

(在私) 任 排 应

大整数乘法 和矩阵乘法

- int型变量的存储空间为32位, 因此变量的取值范围在- $2^{31}\sim 2^{31}(-2,147,483,648\sim 2,147,483,647)$ 之间
- long型 变 量 的 存 储 空 间 为64位, 因 此 变 量 的 取 值 范 围 在- $2^{63}$ ~ $2^{63}$ (-9,223,372,036,854,775,808~9,223,372,036,854,775,807) 之间
- ●两个100位的十进制数相乘,能否使用整数int型或,long型表示?
- 不行,10<sup>100</sup> >> 2<sup>63</sup>

分治法

讲授者 王爱 娟

目录

递归分析: 法

快速排序 堆和堆排序

大整数乘法 和矩阵乘法

- int型变量的存储空间为32位, 因此变量的取值范围在- $2^{31} \sim 2^{31}(-2.147.483.648 \sim 2.147.483.647)$ 之间
- long型 变 量 的 存 储 空 间 为64位, 因 此 变 量 的 取 值 范 围 在-2<sup>63</sup>~2<sup>63</sup>(-9,223,372,036,854,775,808~9,223,372,036,854,775,807) 之间
- ●两个100位的十进制数相乘,能否使用整数int型或,long型表示?
- 不行, $10^{100} >> 2^{63}$

分治法

讲授者 王爱 娟

目录

法 分治法的思

合并排序 快速排序 堆和堆排

大整数乘法 和矩阵乘法

- int型变量的存储空间为32位, 因此变量的取值范围在- $2^{31} \sim 2^{31}(-2.147,483.648 \sim 2.147,483.647)$ 之间
- long型 变 量 的 存 储 空 间 为64位, 因 此 变 量 的 取 值 范 围 在-2<sup>63</sup>~2<sup>63</sup>(-9,223,372,036,854,775,808~9,223,372,036,854,775,807) 之间
- •两个100位的十进制数相乘,能否使用整数int型或,long型表示?
- $\pi$ 6,  $10^{100} >> 2^{63}$

分治法

讲授者 王愛 娟

目录

法 分治法的思 想

快速排序堆和堆排序

- int型变量的存储空间为32位, 因此变量的取值范围在- $2^{31} \sim 2^{31}(-2.147,483.648 \sim 2.147,483.647)$ 之间
- long型 变 量 的 存 储 空 间 为64位, 因 此 变 量 的 取 值 范 围 在-2<sup>63</sup>~2<sup>63</sup>(-9,223,372,036,854,775,808~9,223,372,036,854,775,807) 之间
- ●两个100位的十进制数相乘,能否使用整数int型或,long型表示?
- 不行, 10<sup>100</sup> >> 2<sup>63</sup>



分治法

讲授者 王爱 娟

目录

归分析:

分治法的思 <sub>相</sub>

合并排序快速排序

大整数乘法

#### 大整数乘法问题

- 某些应用中,如当代的密码技术,需要计算超过上千位的 二进制的乘法;
- 假设X和Y是两个n位的二进制数, $n=2^k$ ,计算XY.

#### 解决办法

- 按照通常的做法,需要总共n²次乘法计算;
- 考虑分治法,将X和Y分成相等的两段,每段n/2位,即

$$X = A2^{n/2} + B$$

$$Y = C2^{n/2} + D$$

$$XY = AC2^{n} + (AD + BC)2^{n/2} + BD$$

• 规模为n的原问题转换为4个规模为n/2的子问题;

分治法

讲授者 王爱 娟

目录

法 分治法的思

合并排序 快速排序

大整数乘法

#### 分治法求解

复杂度分析:

• 计算需要的乘法次数递归方程:

$$W(n) = \begin{cases} 4W(n/2), & n > 1 \\ 1, & n = 1 \end{cases}$$

• 根据主定理可得到 $W(n) \in \Theta(n^2)$ .

#### 注意:

- 虽然采用了分治法, 但是时间复杂度并没有降低;
- 回顾主定理, 当问题规模减半b=2, 合并(加减法运算次数)的复杂度为 $\Theta(n)$ 时, 子问题数a>2 时, 时间复杂度为 $\Theta(n^{log_ba})$ , 减少子问题数可降低时间复杂度。

分治法

讲授者 王爱 娟

目录

归分析方

分治法的思 想

合并排序 快速排序

堆和堆排)

大整数乘法 和矩阵乘法

#### 改进的分治法

改进思路:

- AD + BC = (A B)(D C) + AC + BD
- AC和BD已知,那么子问题数目从原来的4个变为3个;

#### 改进的分治法

改进思路:

- AD + BC = (A B)(D C) + AC + BD
- AC和BD已知, 那么子问题数目从原来的4个变为3个;

#### 时间复杂度分析

可记为cn:

- 乘法次数递归方程: M(n) = 3M(n/2), n > 1; M(1) = 1;
- 根据主定理可得到 $M(n) \in \Theta(n^{\log_2 3}) \approx (n^{1.59});$
- 合并子问题的复杂度:6次规模为n的整数加减法操作。
- 加减次数递归方程: A(n) = 3A(n/2) + cn, n > 1; A(1) =0:
  - 根据主定理可得到 $A(n) \in \Theta(n^{\log_2 3}) \approx (n^{1.59})$ ;
- 算法总的时间效率为 $\Theta(n^{1.59})$ , 效率有明显的提升;

- 讲授者 王夢

# 矩阵乘法

分治法 讲授者 王愛

目录 递归分析方

分治法的思想 想 会并排序

唯和堆排序 大整数乘法 和矩阵乘法

#### 矩阵相乘问题

假设 $A \rightarrow B$ 是两个n阶的矩阵,  $n = 2^k$ , 计算C = AB;

#### 解决办法

- $C_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$
- 计算 $C_{ij}$ 需要n次乘法(不考虑加法), 计算C需要 $n^3$ 次乘法:
  - 考虑分治法:

$$\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right] \left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_{22} \end{array}\right] = \left[\begin{array}{cc} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array}\right]$$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}, C_{12} = A_{11}B_{12} + A_{12}B_{22}$$
  
 $C_{21} = A_{21}B_{11} + A_{22}B_{21}, C_{22} = A_{21}B_{12} + A_{22}B_{22}$ 

• 规模为n的原问题转换为8个规模为n/2的子问题:

# 矩阵乘法

分治法

讲授者 王爱 娟

目录

归分析:

分治法的。 <sub>租</sub>

合并排序 快速排序

惟和堆排序

大整数乘法 和矩阵乘法

#### 分治法求解

复杂度分析

• 乘法次数递归方程:

$$M\left(n\right) = \left\{ \begin{array}{l} 8M\left(n/2\right), \ n > 1 \\ 1, \ n = 1 \end{array} \right.$$

• 根据主定理可得到 $M(n) \in \Theta(n^{log8}) = \Theta(n^3)$ 

#### 注意:

- 与传统方法比, 时间复杂度并没有降低;
- 同样地,考虑减少子问题数以降低时间复杂度。

# Strassen (施特拉森)矩阵乘法

分治:

目录 递归分析方 法

分治法的思 想 合并排序

堆和堆排序 大整数乘法 和矩阵乘法

#### Strassen (施特拉森)矩阵乘法

• 分块矩阵相乘:

$$\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right] \left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_{22} \end{array}\right] = \left[\begin{array}{cc} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array}\right]$$

• 中间结果:

$$M_1 = A_{11}(B_{12} - B_{22}), M_2 = (A_{11} + B_{12})B_{22}$$

$$M_3 = (A_{21} + B_{22})B_{11}, M_4 = A_{22}(B_{21} - B_{11})$$

$$M_5 = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$M_6 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$M_7 = (A_{11} - A_{21})(B_{11} + B_{12})$$

• 计算最终结果:

$$C_{11} = M_5 + M_4 - M_2 + M_6$$

$$C_{12} = M_1 + M_2, C_{21} = M_3 + M_4$$

$$C_{22} = M_5 + M_1 - M_3 - M_7$$

# Strassen (施特拉森)矩阵乘法

分治法 讲损者 王憂

录

分治法的思 <sub>相</sub>

> ·并排序 :速排序

大整数乘法 和矩阵乘法

## 时间复杂度分析

- 合并子问题的复杂度:矩阵加法 ((n/2)²个元素相加) 18次;
- 乘法计算总次数递归方程:

$$M(n) = \begin{cases} 7M(n/2), & n > 1 \\ 1, & n = 1 \end{cases}$$

- 根据主定理可得到 $M(n) \in \Theta(n^{log_27}) \approx (n^{2.807})$
- 加法计算总次数递归方程:

$$A(n) = \begin{cases} 7A(n/2) + 18(n/2)^2, & n > 1 \\ 0, & n = 1 \end{cases}$$

- 根据主定理可得到 $A(n) \in \Theta(n^{log_27}) \approx \Theta(n^{2.807})$
- 算法总的时间效率为 $\Theta(n^{2.807})$ ,效率有明显的提升;

分治法

讲授者 王爱 娟

目录

归分析え

分治法的) <sub>租</sub>

想人工证点

中月 和7月 快速排序

Developing

堆和堆排序

大整数乘法 和矩阵乘法 (1) 习题5.1: 8,9

(2) 习题5.4: 7

(3) 习题5.2: **11** (查找教材最后的英文文献[Raw91], 阅读相关内容)