Design and Analysis of Algorithms

Lecture 8: Selection Problem

童咏昕

北京航空航天大学
计算机学院

问题背景：最小值查找

- 给定数组$A[1..16]$，寻找其中最小值

21	17	37	28	13	14	22	52	40	24	48	4	47	8	42	18

- 依次扫描，记录最小值

21	17	37	28	13	14	22	52	40	24	48	4	47	8	42	18

扫描

问题：如何求得数组中第k小的元素？

问题定义

- ## 形式化定义

次序选择问题

Selection Problem

输入

- 包含n个不同元素的数组$A[1..n]$
- 整数$k(1 \le k \le n)$

输出

- 数组$A[1..n]$中第k小的元素$(1 \le k \le n)$

排序求解

- 数组排序
  - 求得所有元素的次序
  - 时间复杂度：$O(n\log n)$
- 选择元素
  - 求得第8小的元素
  - 时间复杂度：$O(1)$

21	17	37	28	13	14	22	52	40	24	48	4	47	8	42	18

↓ 排序

查找第8小元素

4	8	13	14	17	18	21	22	24	28	37	40	42	47	48	52

问题分析

- 数组排序
  - 求得所有元素的次序
  - 时间复杂度：$O(n\log n)$

					第6小元素		第8小元素							第15小元素	
4	8	13	14	17	18	21	22	24	28	37	40	42	47	48	52

问题：是否有必要求得所有元素的次序？

问题分析

- 次序选择
  - 不必求得所有元素次序
  - 时间复杂度：$O(?)$

第6小元素　第8小元素　第15小元素

4	8	13	14	17	18	21	22	24	28	37	40	42	47	48	52

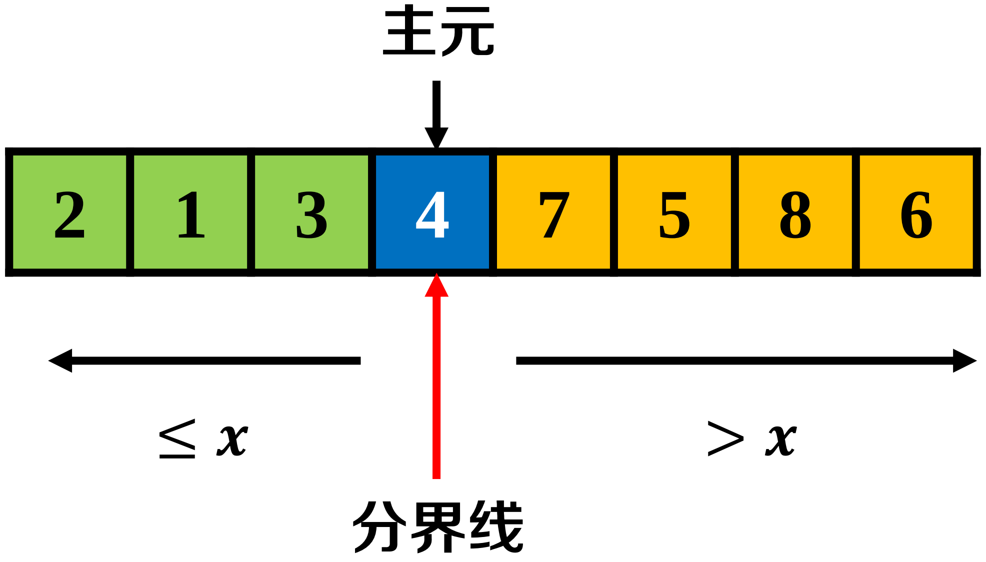
查找第8小元素

17	13	14	4	8	18	21	22	28	24	37	40	47	52	42	48

受启发于快速排序的数组划分

回顾与启发

- 选取固定位置主元x（如尾元素）
- 维护两个部分的右端点变量i, j
- 考察数组元素$A[j]$，只和主元比较
  - 若$A[j] \leq x$，则交换$A[j]$和$A[i+1]$，i与j右移
  - 若$A[j] > x$，则j右移



固定位置划分求解

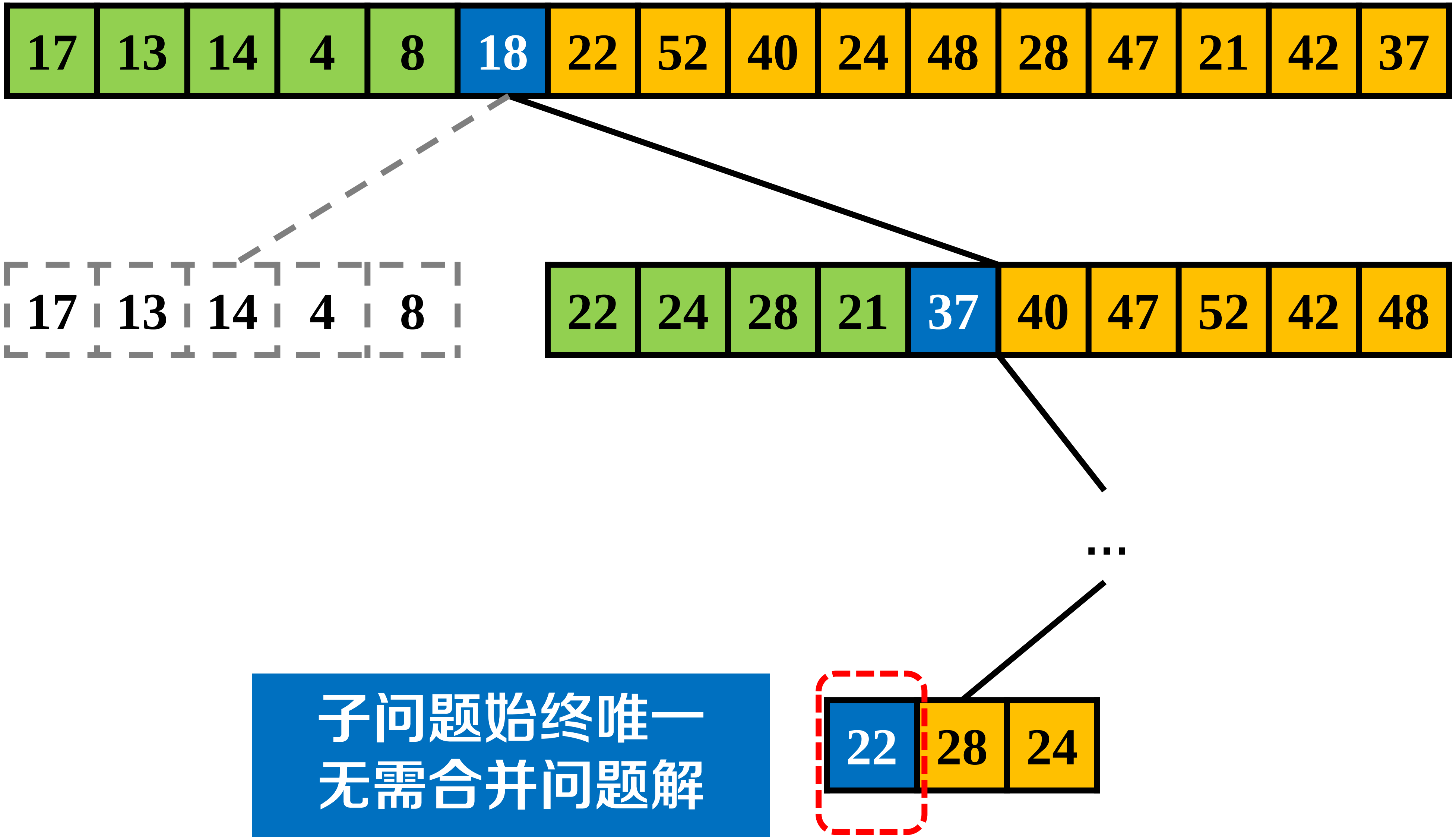
- 选取固定位置主元，小于主元的元素个数$q-p$
  - 情况1：$k=q-p+1$，$A[q]$为数组第k小元素
  - 情况2：$k<q-p+1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k>q-p+1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素

p					q										r
17	13	14	4	8	18	22	52	40	24	48	28	47	21	42	37

$q-p+1$ $\quad$ $k-(q-p+1)$

17	13	14	4	8	18	22	52	40	24	48	28	47	21	42	37

固定位置划分求解



子问题始终唯一
无需合并问题解

分而治之框架

- 分解原问题
- 解决子问题
- 合并问题解

算法实例

- 选取固定位置主元
  - 情况1：$k = q - p + 1$，$A[q]$为数组第k小元素
  - 情况2：$k < q - p + 1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k > q - p + 1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素

$p = 1$

21	17	37	28	13	14	22	52	40	24	48	4	47	8	42	18

$r = 16$

查找第8小元素

算法实例

- 选取固定位置主元
  - 情况1：$k = q - p + 1$，$A[q]$为数组第k小元素
  - 情况2：$k < q - p + 1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k > q - p + 1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素

$p=1$	17	13	14	4	8	18	22	52	40	24	48	28	47	21	42	37	$r=16$
						主元											

查找第8小元素

$q = 6$

$q - p + 1 = 6 - 1 + 1 = 6$

算法实例

- 选取固定位置主元
  - 情况1：$k = q - p + 1$，$A[q]$为数组第k小元素
  - 情况2：$k < q - p + 1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k > q - p + 1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素

17	13	14	4	8	18	22	52	40	24	48	28	47	21	42	37

$p = 7$

22	24	28	21	37	40	47	52	42	48

$r = 16$

主元

$q = 11$

查找第2小元素

$q - p + 1 = 11 - 7 + 1 = 5$

算法实例

- **选取固定位置主元**
  - 情况1：$k = q - p + 1$，$A[q]$为数组第k小元素
  - 情况2：$k < q - p + 1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k > q - p + 1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素

17	13	14	4	8	18	22	52	40	24	48	28	47	21	42	37

22	24	28	21	37	40	47	52	42	48

$p = 7$

21	24	28	22

$r = 10$

主元

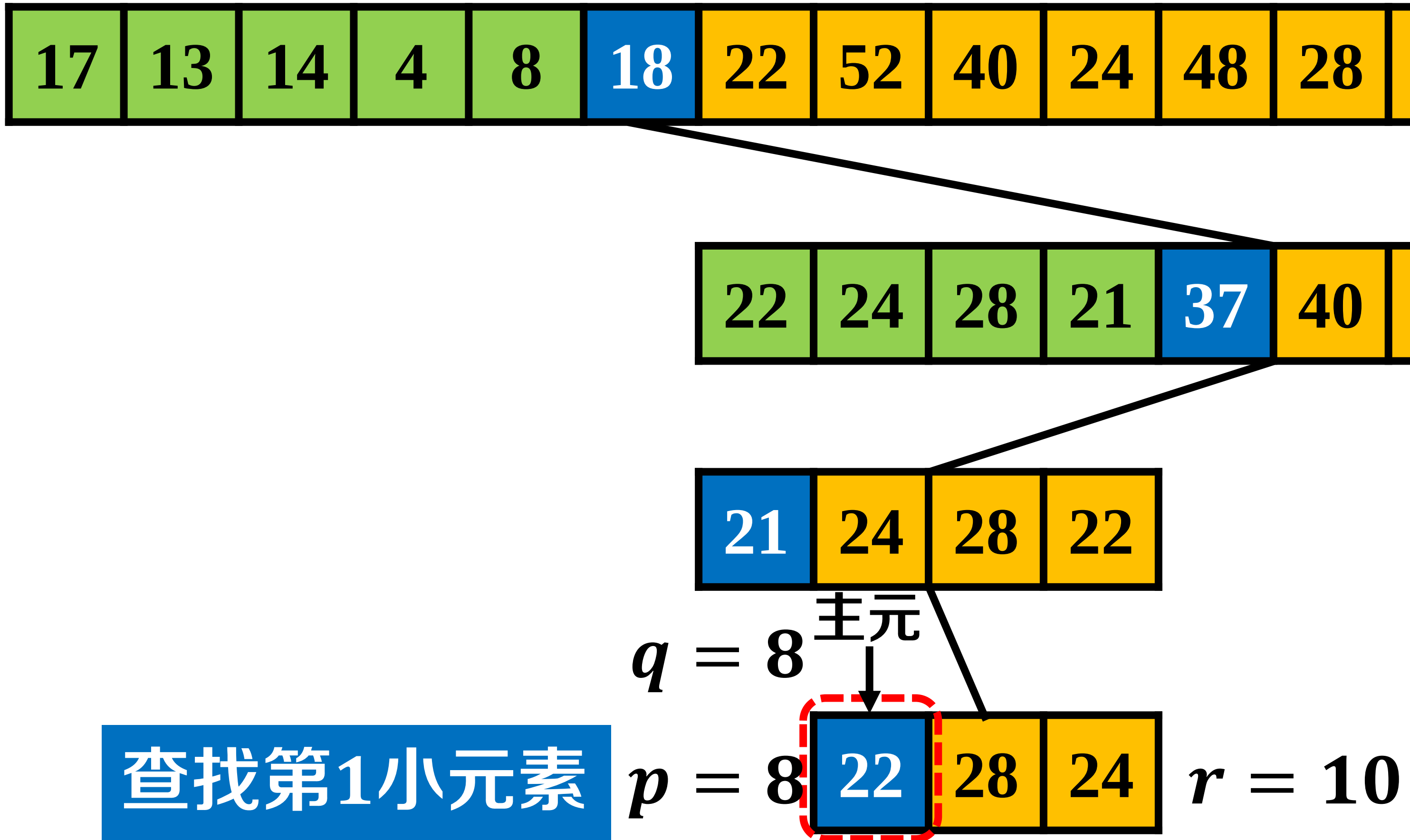
查找第2小元素

$q - p + 1 = 7 - 7 + 1 = 1$

$q = 7$

算法实例

- 选取固定位置主元
  - 情况1：$k = q - p + 1$，$A[q]$为数组第k小元素
  - 情况2：$k < q - p + 1$，在$A[p..q-1]$中寻找第k小元素
  - 情况3：$k > q - p + 1$，在$A[q+1..r]$中寻找第$k-(q-p+1)$小元素



固定位置划分：伪代码

- **Partition(A, p, r)**

```
输入: 数组A, 起始位置p, 终止位置r
输出: 划分位置q
x ← A[r]
i ← p − 1
for j ← p to r − 1 do
    if A[j] ≤ x then
        exchange A[i + 1] with A[j]
        i ← i + 1
    end
end
exchange A[i + 1] with A[r]
q ← i + 1
return q
```

固定位置次序选择：伪代码

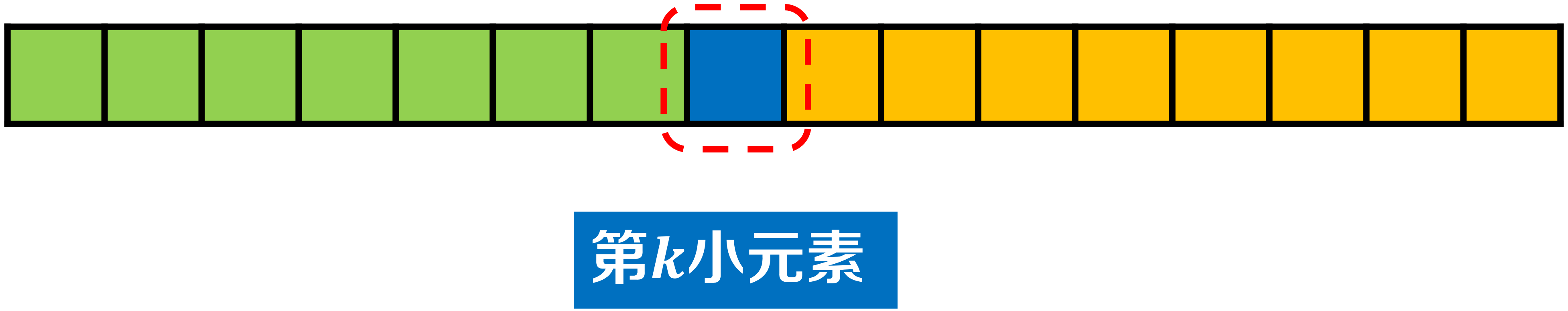
- **Selection($\boldsymbol{A, p, r, k}$)**

输入: 数组A,起始位置p,终止位置r,元素次序k
输出: 第k小元素x

$q \leftarrow \text{Partition(A, p, r)}$
if $k = (q - p + 1)$ **then**
$\quad x \leftarrow A[q]$
end
if $k < (q - p + 1)$ **then**
$\quad x \leftarrow \text{Selection}(A, p, q - 1, k)$
end
if $k > (q - p + 1)$ **then**
$\quad x \leftarrow \text{Selection}(A, q + 1, r, k - (q - p + 1))$
end
return x

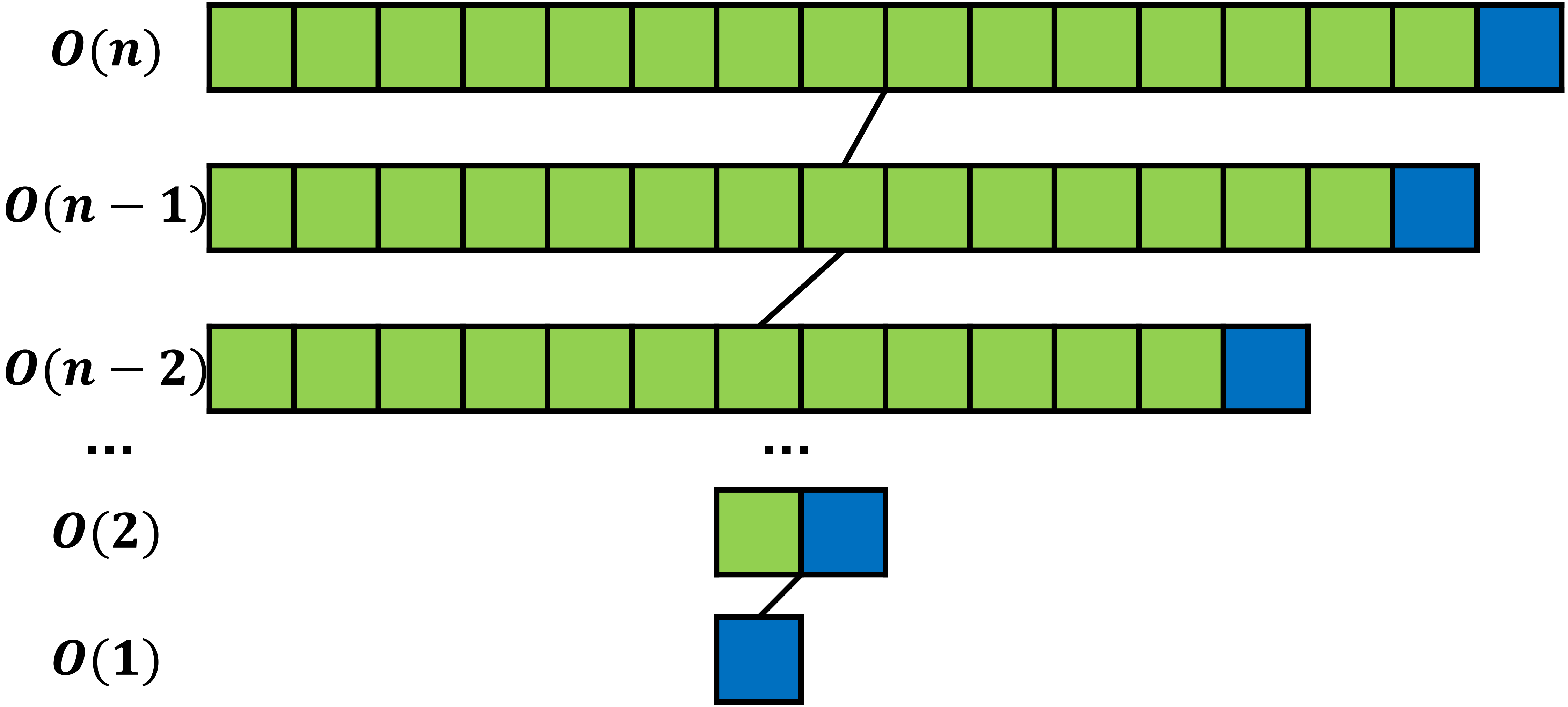
复杂度分析：最好情况

$O(n)$



- 时间复杂度：$T(n) = O(n)$

复杂度分析：最坏情况

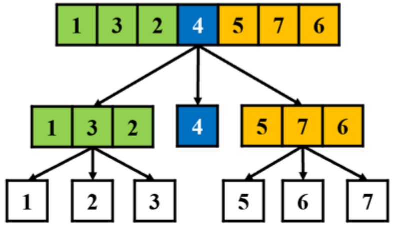


- 时间复杂度：$T(n) = \sum_{i=1}^{n} i \leq n^2 = O(n^2)$

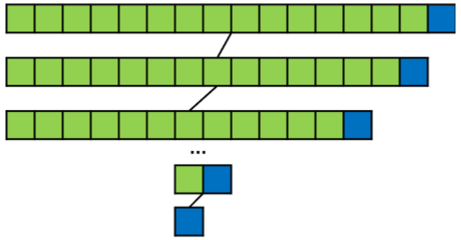
复杂度比较

算法名称	最好情况复杂度	最坏情况复杂度
固定位置快速排序	$O(n\log n)$	$O(n^2)$
固定位置次序选择	$O(n)$	$O(n^2)$

快速排序



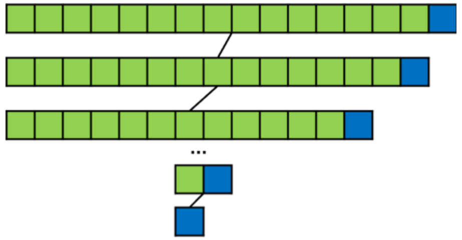
最好情况：$O(n\log n)$



最坏情况：$O(n^2)$

次序选择

最好情况：$O(n)$



最坏情况：$O(n^2)$

复杂度比较

算法名称	最好情况复杂度	最坏情况复杂度
固定位置快速排序	$O(n\log n)$	$O(n^2)$
固定位置次序选择	$O(n)$	$O(n^2)$

问题：如何摆脱最坏情况的困境?

使用**随机位置**划分

随机位置划分：伪代码

- **Randomized-Partition(A, p, r)**

输入: 数组A, 起始位置p, 终止位置r
输出: 划分位置q

$s \leftarrow \text{Random}(\text{p}, \text{r})$
exchange $A[s]$ with $A[r]$
$q \leftarrow \text{Partition}(A, p, r)$
return q

随机选择主元

随机位置次序选择：伪代码

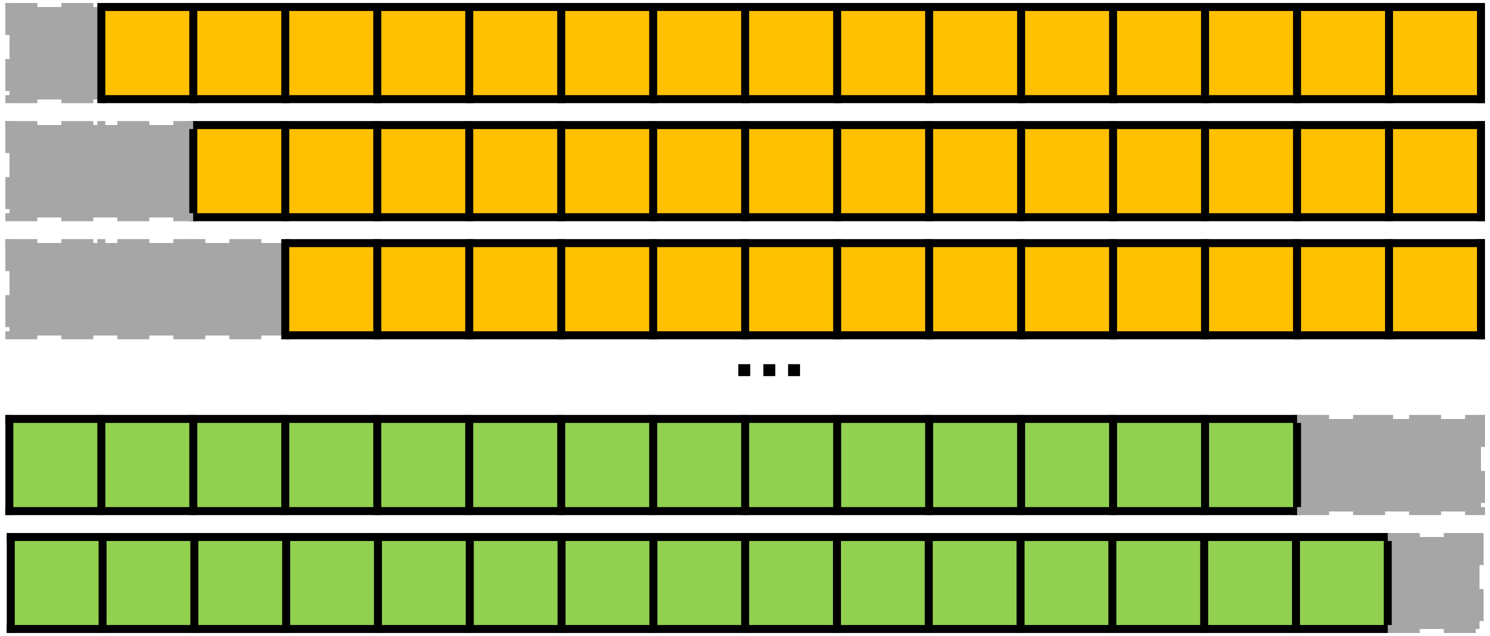
- **Randomized-Selection(A, p, r, k)**

输入: 数组A,起始位置p,终止位置r,元素次序k
输出: 第k小元素x

$q \leftarrow \text{Randomized-Partition}(A, p, r)$　　随机划分数组
if $k = (q - p + 1)$ **then**
  $x \leftarrow A[q]$
end
if $k < (q - p + 1)$ **then**
  $x \leftarrow \text{Randomized-Selection}(A, p, q - 1, k)$
end
if $k > (q - p + 1)$ **then**
  $x \leftarrow \text{Randomized-Selection}(A, q + 1, r, k - (q - p + 1))$
end
return x

复杂度分析：期望情况

- 随机选择主元，共n种情况



$0：n-1$

$1：n-2$

$2：n-3$

…

$n-2：1$

$n-1：0$

- $T(n) \le \begin{cases} \color{red}{T(n-1)} + O(n) \\ T(n-2) + O(n) \\ T(n-3) + O(n) \\ \dots \\ \color{red}{T(n-1)} + O(n) \end{cases}$

n种情况概率均为$1/n$

每个值$T(i)$出现2次，$i \ge \left\lfloor \frac{n}{2} \right\rfloor$

复杂度分析：期望情况

- **期望时间：**

$$E[T(n)] \le E[\frac{2}{n} \cdot \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} \left(T(i) + O(n)\right)]$$

$$\le \frac{2}{n} \cdot \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(i)] + \frac{2}{n} \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} O(n)$$

$$\le \frac{2}{n} \cdot \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(i)] + O(n)$$

问题：如何进一步求解该递归式？

复杂度分析：期望情况

算法名称	最好时间复杂度	最坏时间复杂度	期望时间复杂度
快速排序	$O(n\log n)$	$O(n^2)$	$O(n\log n)$
次序选择	$O(n)$	$O(n^2)$	$O(n)$

问题：次序选择期望时间复杂度是否为$O(n)$?

使用代入法验证

代入法分析：期望情况

- 最好情况：$T(n) = O(n)$
- 假设：$\forall i < n,\ E[T(i)] \le c \cdot i$

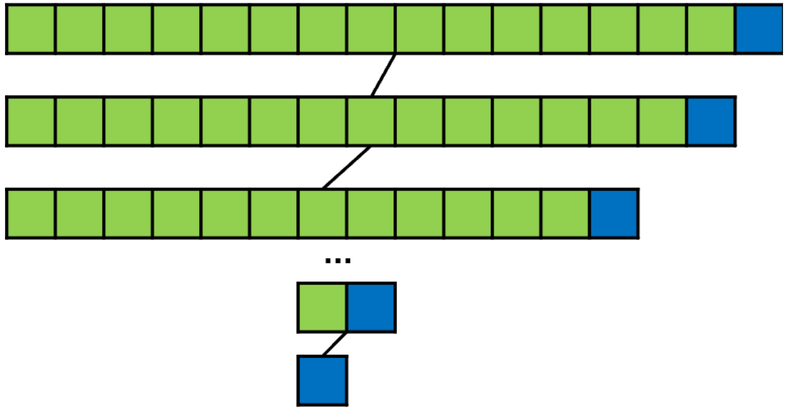
$$
\begin{aligned}
E[T(n)] &\le O(n) + \frac{2}{n} \cdot \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(i)] \\
&\le O(n) + \frac{2}{n} \cdot \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n-1} c \cdot i \\
&\le O(n) + \frac{2}{n} \cdot c \cdot \frac{3}{8} n^2 \\
&= c \cdot n - \left(\frac{1}{4} c \cdot n - O(n) \right) \\
&\le c \cdot n
\end{aligned}
$$

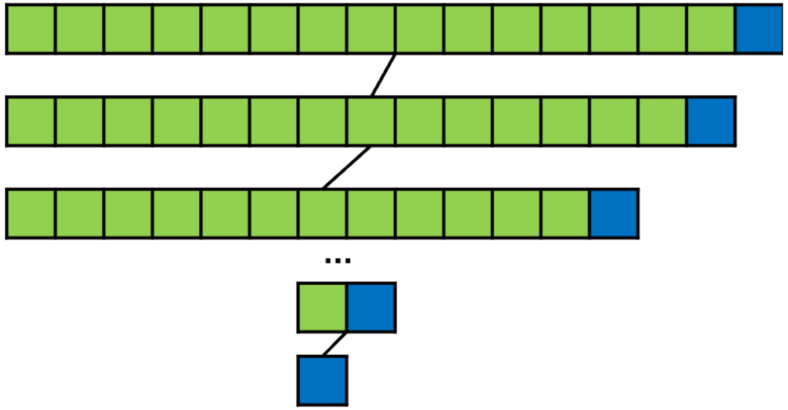
随机位置次序选择：期望时间复杂度为$O(n)$

小结

快速排序　　次序选择

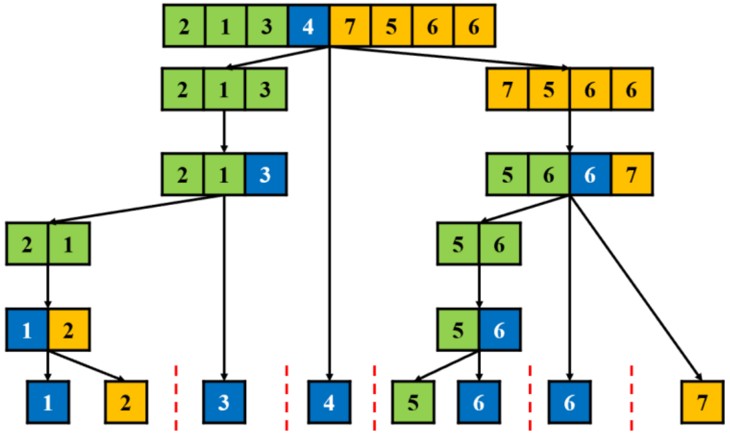
固定位置划分



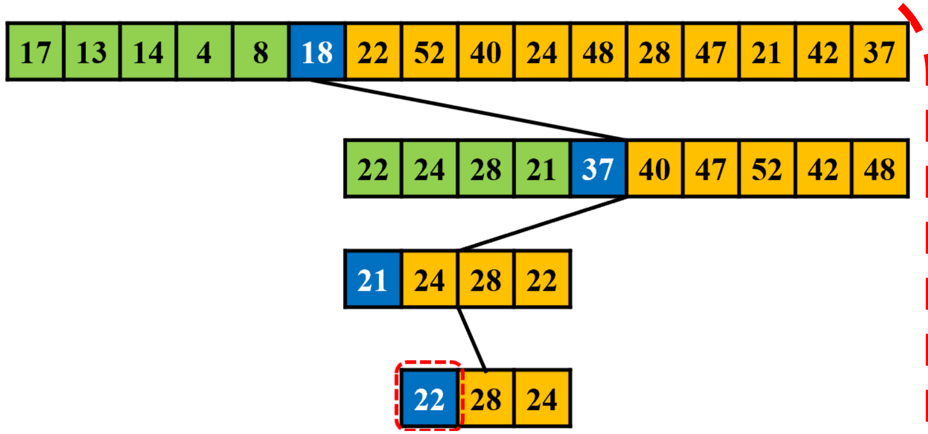


最坏情况：$O(n^2)$　　最坏情况：$O(n^2)$

随机位置划分



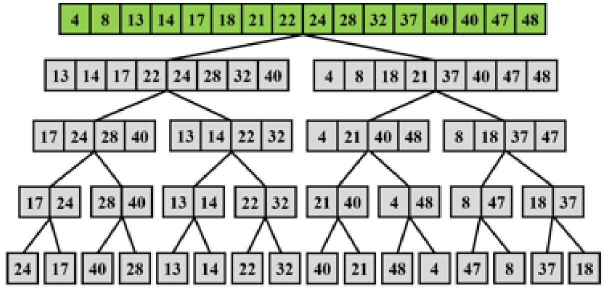
期望情况：$O(n\log n)$



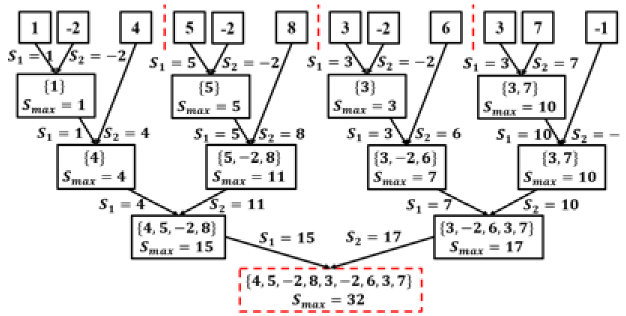
期望情况：$O(n)$

总结

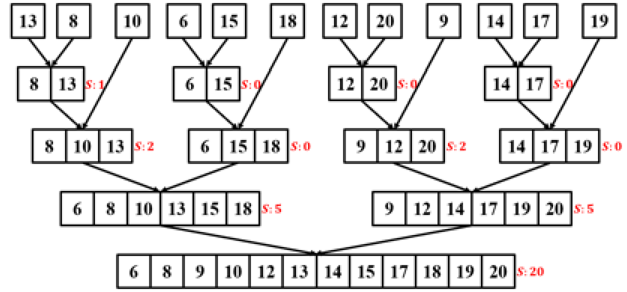
归并排序



最大子数组



逆序计数



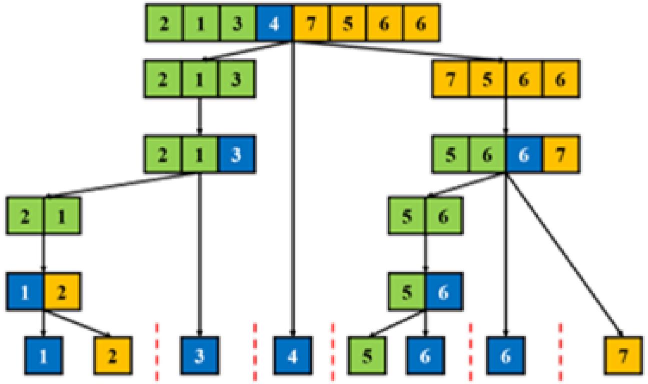
分而治之框架

分解原问题

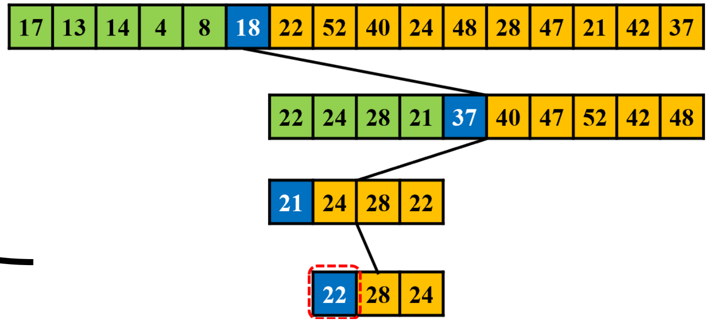
解决子问题

合并问题解

快速排序



次序选择



谢谢

BDA

Beihang University