第6章　分治策略

林 海
Lin.hai@whu.edu.cn

分治法

- **To solve P:**

  -- 分解 P into smaller problems P_1, P_2, …, P_k.

  -- 解决 by solving the (smaller) subproblems recursively.

  -- 合并 the solutions to P_1, P_2, …, P_k into the solution for P.

由于子问题与原问题是同类的,故分治法
可以很自然地应用递归。

分治算法形式

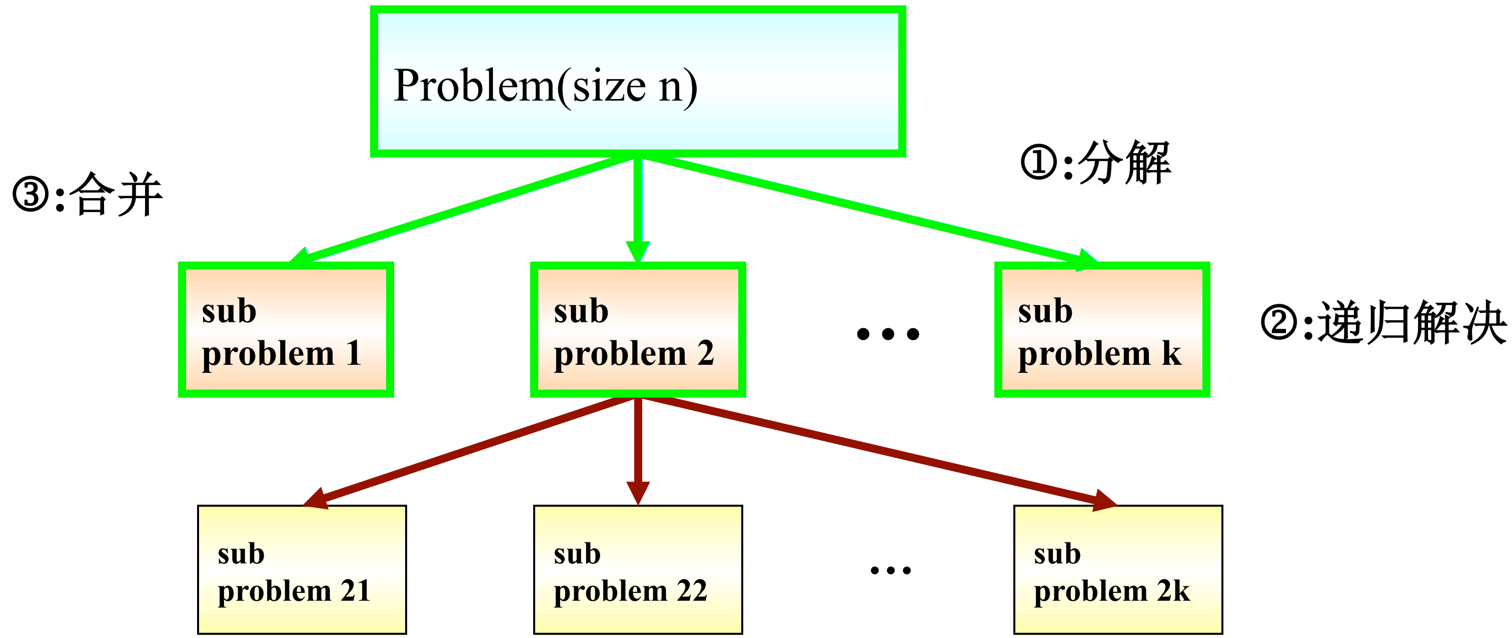
(1) 如果实例 I 的规模是“小”的，则使用直接的方法求解问题并返回其答案，否则继续做下一步。

(2) 把实例 I 分割成 p 个大小几乎相同的子实例 $I_1, I_2, \cdots, I_p$。

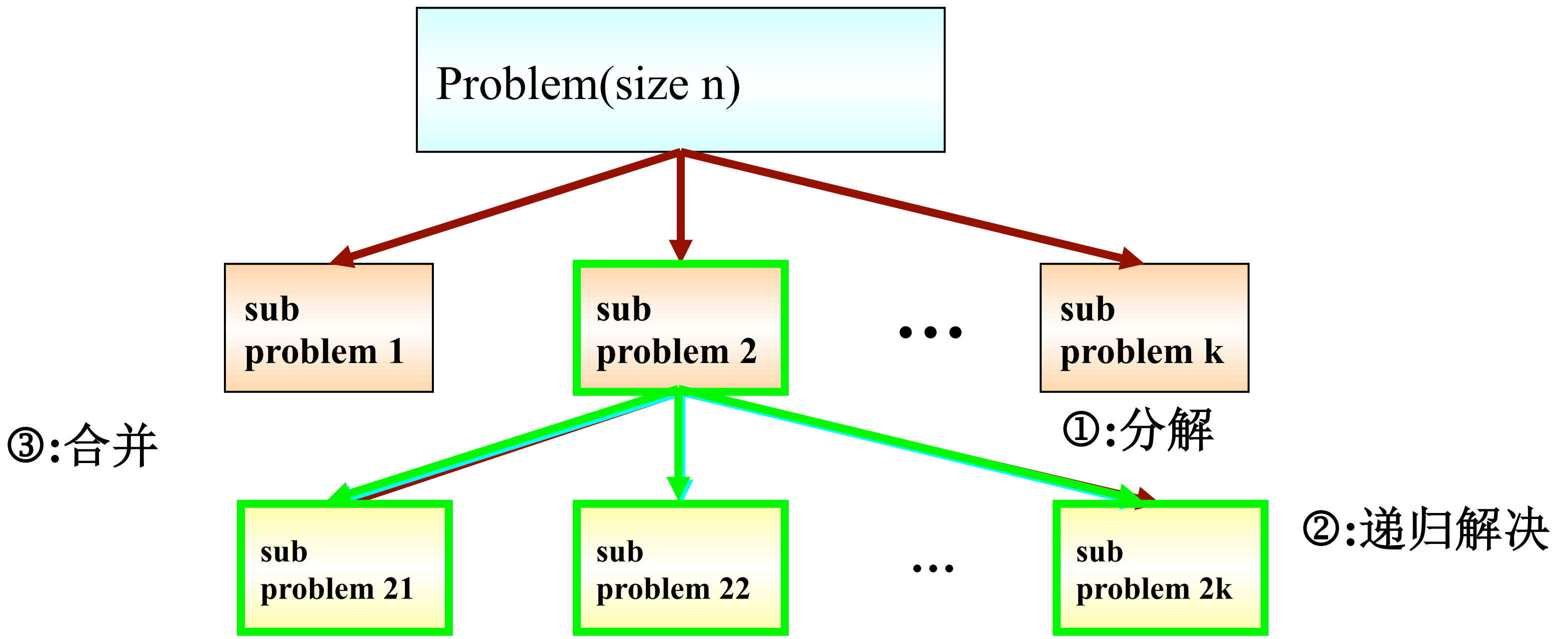
(3) 对每个子实例 I_j，$1 \leqslant j \leqslant p$，递归调用算法，并得到 p 个部分解。

(4) 组合 p 个部分解的结果得到原实例 I 的解，返回实例 I 的解。

分治法


Problem(size n)
③:合并
①:分解
sub problem 1
sub problem 2
• • •
sub problem k
②:递归解决
sub problem 21
sub problem 22
• • •
sub problem 2k

分治法

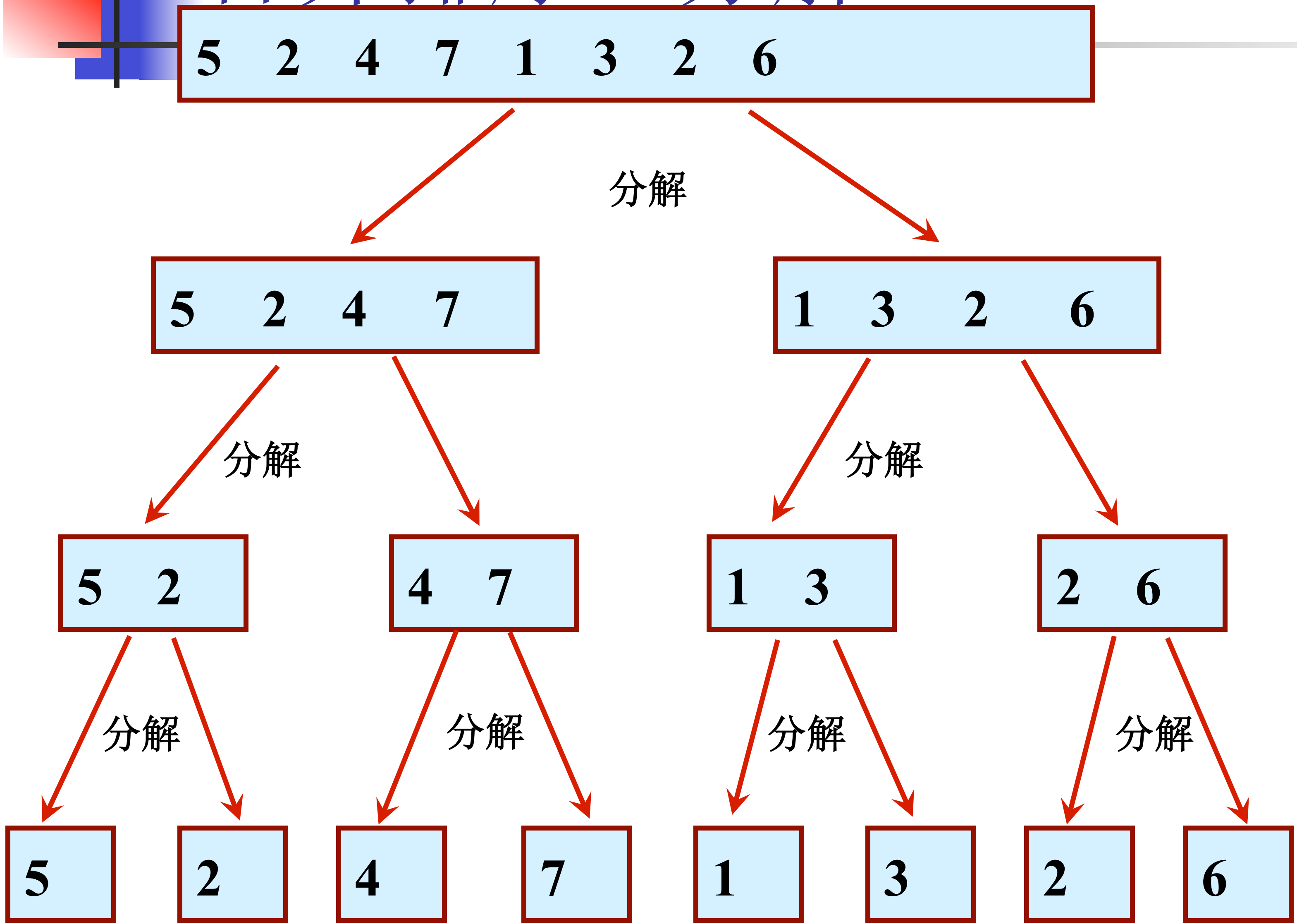


Problem(size n)
sub problem 1
sub problem 2
…
sub problem k
①:分解
③:合并
sub problem 21
sub problem 22
…
sub problem 2k
②:递归解决

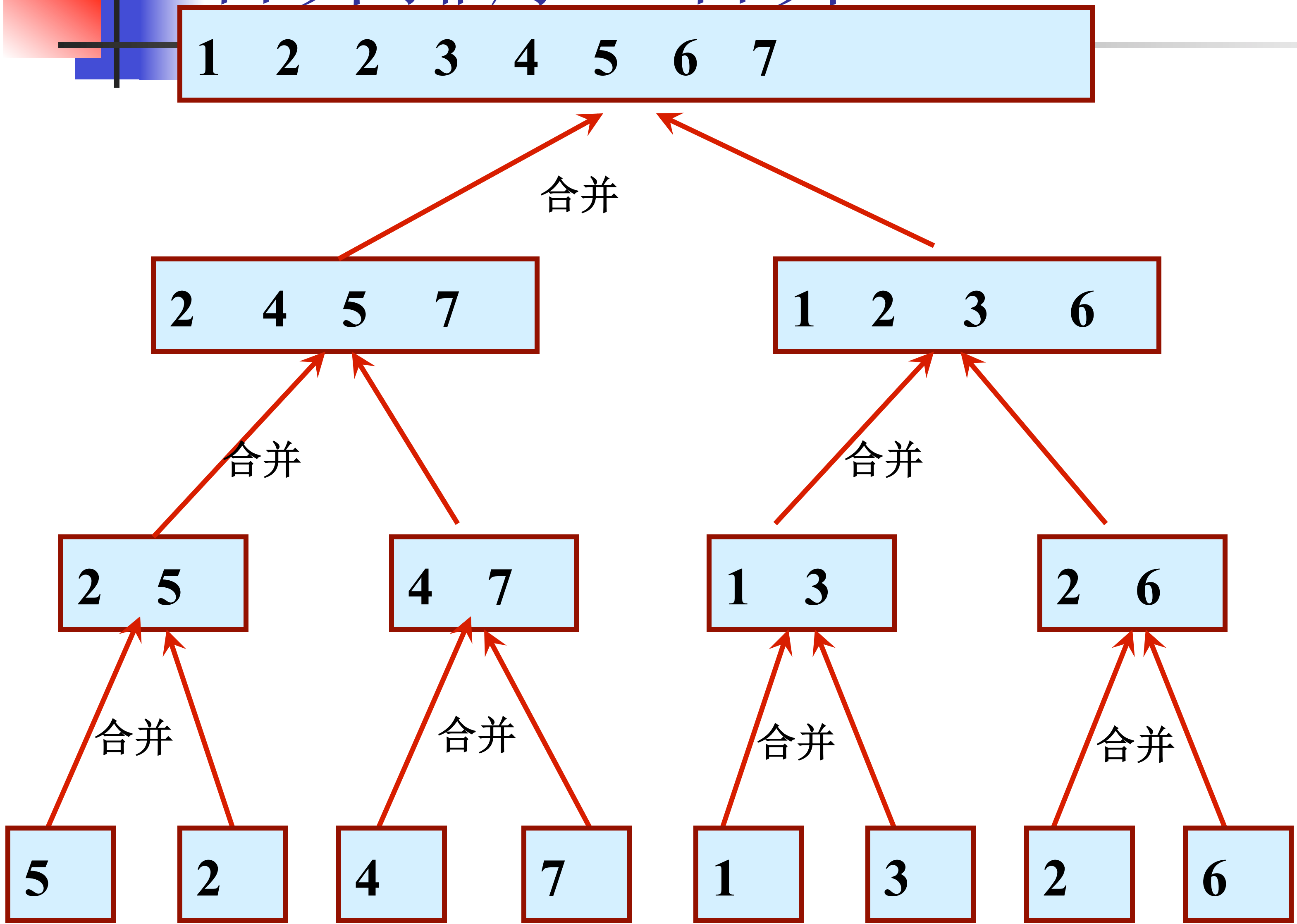
6.3 合并排序

- **Using 分解-解决-合并， we can obtain a merge-sort algorithm**
  -- 分解: 将n个元素的数组分成两个元素个数为n/2的子数组.
  -- 解决: 递归的解决子数组.
  -- 合并: 将两个排序好的子数组合并成一个数组.

合并排序—分解



合并排序—合并



算法分析

- Merge的复杂度为 $\Theta(n)$
- 合并排序的复杂度为

分解：分解步骤仅仅计算子数组的中间位置，需要常量时间，因此，$D(n)=\Theta(1)$。

解决：我们递归地求解两个规模均为 $n/2$ 的子问题，将贡献 $2T(n/2)$的运行时间。

合并：我们已经注意到在一个具有 n 个元素的子数组上过程 MERGE 需要 $\Theta(n)$的时间，所以 $C(n)=\Theta(n)$。

$$T(n)=\begin{cases}\Theta(1) & 若\ n=1\\ 2T(n/2)+\Theta(n) & 若\ n>1\end{cases}$$

解此方程得出复杂度为 $\Theta(n\lg n)$

6.5 寻找中项和第k小元素

- 给定已排好序(非降序)的数组A[1…n],中项是指其“中间”元素。若n为奇数，则中项为数组中的第(n+1)/2个元素；若n为偶数，则存在两个中间元素，分别为第n/2和第n/2+1个元素，在这种情形下，我们取第n/2个元素作为中项；综上，中项为第$\lceil n/2 \rceil$个最小元素。
- 寻找中项的一个直接的方法：先排序，后取中项。显然，该方法的时间复杂度至少为Ω(nlogn)。能否找到更为高效的方法?
- 寻找中项是寻找第k小元素的一个特列。如果能解决寻找第k小元素的问题，那么当k=$\lceil n/2 \rceil$时，解决的就是寻找中项问题。
- 回顾二分搜索：以中间元素为基准抛弃部分元素，不断减小问题规模。

思路

- 怎么舍去一半？
  - 找到中间元素（大概即可）
  - 将n个元素划分成$\lfloor n/5 \rfloor$组，每组5个元素，如果n不是5的倍数，则排除剩余的元素。
  - 对每组元素排序，并取出它们的中项(即第3个元素)。 $\lfloor n/5 \rfloor$个中项的中项，我们记为mm。
- 依据mm将数组A划分为三个子数组：$A_1=\{a|a<mm\}$、$A_2=\{a|a=mm\}$、$A_3=\{a|a>mm\}$。
- 判断第k小元素可能在哪一个子数组中出现：如果在A_2中出现，则已经找到；否在，在A_1或A_3上进行递归。

为方便演示，设阈值为6。现要寻找下面数组A中的第13小元素：

A={8,33,17,51,57,49,35,11,25,37,14,3,2,13,52,12,6,29,32,54,5,16,22,23,7}

{8,33,17,51,57} {49,35,11,25,37} {14,3,2,13,52} {12,6,29,32,54} {5,16,22,23,7}

{8,17,33,51,57} {11,25,35,37,49 } {2, 3,13,14,52} {6, 12,29,32,54} {5,7,16,22,23}

M={33,35,13,29,16} mm=29

因为k=13<$|A_1|$=15

A_1={8,17,11,25,14,3,2,13,12,6,5,16,22,23, 7}

A_2={29} A_3={33,51,57,49,35,37,52,32,54}

{8,17,11,25,14} {3,2,13,12,6} {5,16,22,23,7}

{8,11,14,17,25} {2,3,6,12,13} {5,7,16,22,23}

M={14,6,16} mm=14

$M=\{14,6,16\}$ $mm=14$

$A=\{8,17,11,25,14,3,2,13,12,6,5,16,22,23,7\}$

$A_1=\{8, 11,\cancel{3,2},13,12,6,5,7\}$

$A_2=\cancel{\{14}\}$

$A_3=\{17,25,16,22,23\}$

因为$|A_3|<p=6$ 阈值

因为$k=13>|A_1|+|A_2|=9+1=10$

$\{16,17, 22,23,25,\}$

return A[3] 因为(3=13-9-1)

Algorithm: Select(A[low…high], k)

输入：数组A[low,...high]和整数k，$1\leq k\leq high-low+1$

输出：A[low…high]中的第k小元素

1. $p \leftarrow high-low+1$ //问题的规模 $\Theta(1)$
2. if p <44 then {将 A[low,...high] 排序，return A[low+k-1] // 如果个数小于44，直接排序$\Theta(1)$
3. 令$q=\lfloor p/5\rfloor$。将A[low…high]分成q个子数组，每组5个元素。若5不整除p,则排除剩余的元素。 // $\Theta(n)$
4. 对q个子数组分别进行排序，分别找出中项。这些中项组成一个数组M。 // $\Theta(n)$
5. $mm \leftarrow$ Select(M[1…q], $\lceil q/2 \rceil$) //中项的中项, $\Theta(T\lfloor n/5\rfloor)$
6. 将A[low…high]分成三组

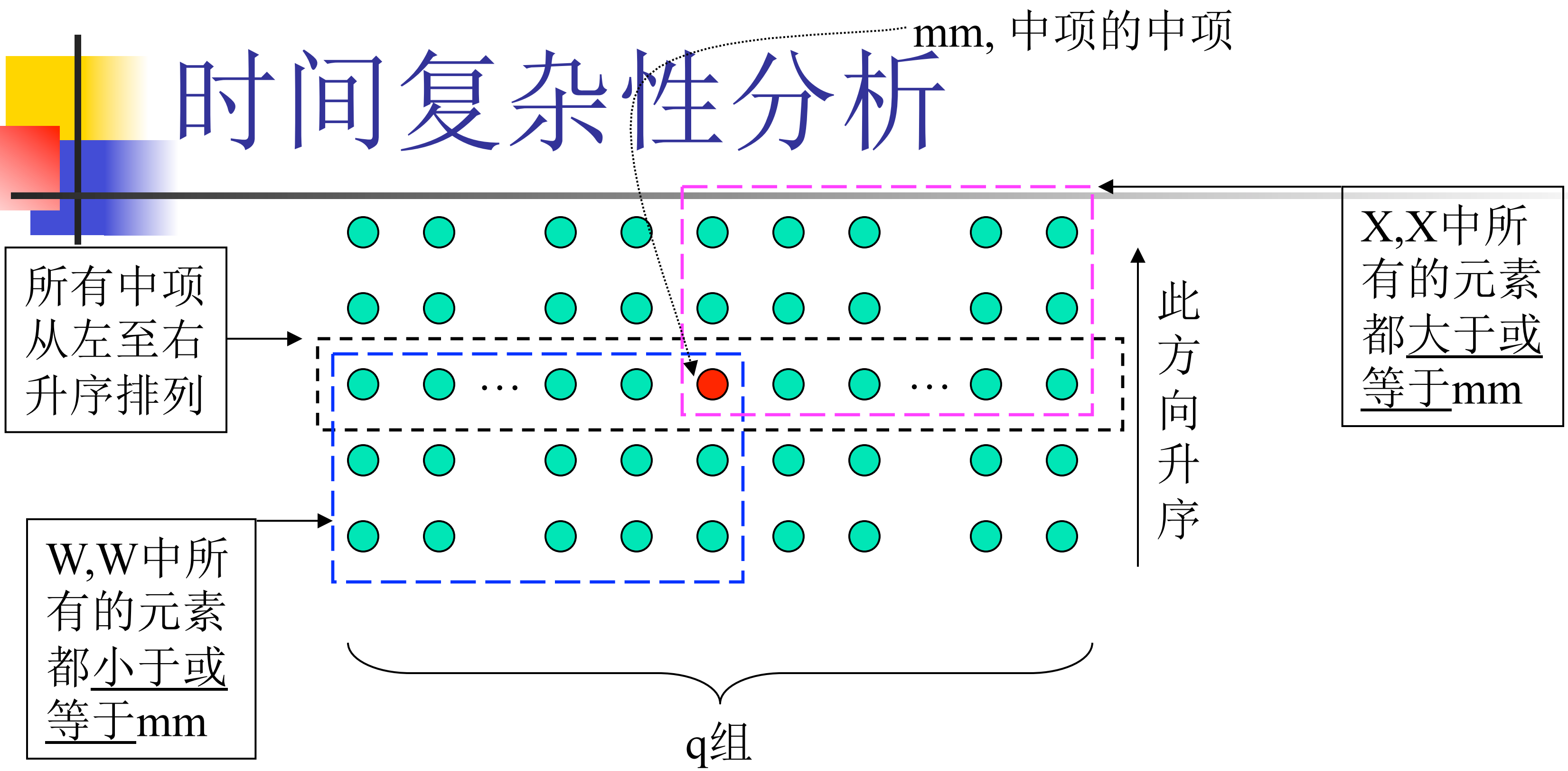
   $A_1=\{a|a<mm\}$ $A_2=\{a|a=mm\}$ $A_3=\{a|a>mm\}$。 //$\Theta(n)$
7. Case //至多T(0.7n+1.2)

   $|A_1|\geq k$: return select($A_1[1\ldots|A_1|]$,k)

   $|A_1|+|A_2|\geq k$: return mm

   $|A_1|+|A_2|<k$: return select($A_3[1,|A_3|]$, $k-|A_1|-|A_2|$)

时间复杂性分析



- A_1^{+}表示A中小于或等于mm的元素集,A_1是A中严格小于mm的元素集。
- A_3^{+}表示A中大于或等于mm的元素集,A_3是A中严格大于mm的元素集。
- 因为 A_1^{+}至少与W同样大(为什么？)，所以$|A_1^{+}| \ge 3\lceil \lfloor n/5\rfloor/2 \rceil \ge 3/2\lfloor n/5\rfloor$，所以，$|A_1| \le \mathbf{n-} 3/2\lfloor n/5\rfloor \le \mathbf{n-} 3/2((n-4)/5) = 0.7n+1.2$
- 由对称性，我们有$|A_3| \le 0.7n+1.2$

- 算法复杂度的递归方程为：
  - 其中T(⌊n/5⌋)为Step 5耗费时间。
  - ST(0.7n+1.2)：下面设法去掉其中的常数1.2。假设0.7n+1.2 ≤⌊0.75n⌋,那么当0.7n+1.2 ≤0.75n-1，即当n ≥44时， 0.7n+1.2 ≤⌊0.75n⌋ 成立。此时，Step 7耗费时间之多为T(⌊0.75n⌋).

$$T(n) \le \begin{cases} c & \text{if } n < 44 \\ T(\lfloor n/5 \rfloor) + T(\lfloor 3n/4 \rfloor) + \Theta(n) & \text{if } n \ge 44 \end{cases}$$

$$\downarrow$$

$$T(n) = \Theta(n)$$

定理2.7