算法分析与设计 第五讲

分治法及相关实例分析（续）

主要内容

- 最大元最小元问题
- 最近点对问题
- 寻找顺序统计量问题

最大元、最小元

- 给定n个数据元素，找出其中的最大元和最小元
  - 直接解法：逐个找，用n-1次比较来找出最大元，再用n-2次比较来找出最小元，比较次数（基本运算）为2n-3次

最大元、最小元

- 分治法
  - 当n=2时，一次比较就可以找出两个数据元素的最大元和最小元
  - 当n>2时，可以把n个数据元素分为大致相等的两半
  - 求数组最大元、最小元的算法下界

$$\lceil 3n/2-2 \rceil$$

最近点对问题

- 对于平面上给定的N个点，给出距离最近的两个点
  - Brute force法：把所有点对逐一检查一遍
    - $T(n)=\Theta(n^2)$
  - 分治策略
    - 如何分解？
    - 如何合并？

一维的最近点对问题

- n个点退化为n个实数，最近点对即为这n个实数中相差最小的两个实数
- 分治法求解
  - 分解：用各点坐标的中位数m作为分割点，分成两个点集
  - 求解：在两个点集上分别找出其最接近点对{p1,p2}和{q1,q2}
  - 合并：整个点集的最近点对或者是{p1,p2}，或者是{q1,q2}，或者是某个{p3,q3}，其中p3和q3分属两个点集

一维的最近点对问题

- 合并
  - 如果最近点对是{p3,q3}，即 |p3-q3|<d，则p3和q3两者与m的距离不超过d，即p3∈(m-d,m]，q3∈(m,m+d]

最近点对问题

- 有n个点，输入点集记为P
- 分解
  - 将P进行分割，分为2部分求最近点对
  - 选择一条垂线L，将P拆分左右两部分为P_L和P_R

最近点对问题

- 解决
  - 分别寻找P_L和P_R中的最近点对及距离，设其找到的最近点对的距离分别是δ_L和 δ_R
  - 置$\delta=\min(\delta_L, \delta_R)$

最近点对问题

- 合并
  - 对于从P_L和P_R求得的δ_L和δ_R，如何合并？
  - 可能一：最近点对就是某次递归调用找出的距离为δ的点对
  - 可能二：最近点对是由P_L 中的一个点和P_R中的一个点组成的点对

最近点对问题

- 合并子问题
  - 关注以直线L为中心，宽度为2δ的垂直带状区域
  - 看是否可以找到一个点对的距离$\delta' < \delta$

最近点对问题

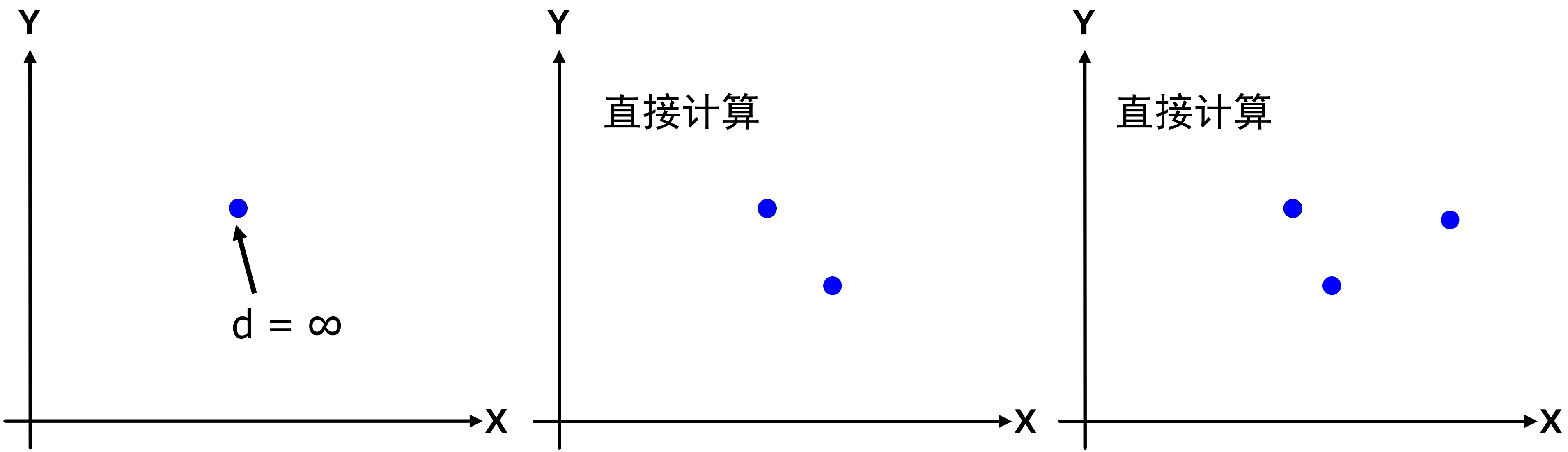
- 合并子问题 考察带状区域中的点
  - 对于带状区域中的每个点p，算法试图找出距离p在δ单位以内的点，如果距离$\delta' < \delta$，则更新当前的最近点对距离
  - 如果带状区域中的点是按y坐标升序排列的，对于每个点p，只需要检查p之后的7个点即可

最近点对问题

- 合并子问题小结
  - 找出以L为中心线，宽度为2δ的带状区域
  - 获得带状区域中排序后的点集Y’
  - 对Y’中的每个点，检查其后面的7个点，计算距离并更新最近点对的距离

最近点对问题

- 递归的出口
- 点数较少时的情形



只有一个点　　只有两个点　　只有三个点

最近点对问题

- 以L为中心线，宽度为2δ的带状区域中的点，如何筛选，并保证有序（按y坐标升序排列）
  - 筛选出来之后按照y坐标递增的方式进行排序？
  - 对Y进行预排序，将排好序的Y传入递归主过程

最近点对问题

- 难点
  - 如何在线性时间内获得

    Y_L, Y_R, X_L, X_R, Y'
  - 若X,Y已按相应坐标排好序

```
length[Y_L] ← length[Y_R] ← 0
for i ← 1 to length[Y]
    do if Y[i] ∈ P_L
          then length[Y_L] ← length[Y_L] + 1
               Y_L[length[Y_L]] ← Y[i]
          else length[Y_R] ← length[Y_R] + 1
               Y_R[length[Y_R]] ← Y[i]
```

最近点对问题

- 分析
- 递归式为

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

$$f(n) = O(n)$$

$$T'(n) = T(n) + O(n\lg n)$$

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

$$T'(n) = O(n\lg n)$$

寻找顺序统计量问题

- 求第i小元素问题、选择问题
  - 设集合S中共有n个数据元素，要在S中找出第i小元素
- 最小元：第1个顺序统计量
- 最大元：第n个顺序统计量
- 中位数： $i = \lfloor (n+1)/2 \rfloor$

寻找顺序统计量问题

- 如下假设：
  - 集合由n个数值不同的元素组成
  - 在此假设下的结论可以推广到有重复数值的情形
- 问题描述：
  - 输入：一个包含n个不同数的集合A和一个数i
  - 输出：元素x，x大于A中其它的i-1个元素

寻找顺序统计量问题

- 求解方法
  - 排序
  - 期望线性时间
  - 最坏情况线性时间

寻找顺序统计量问题

- 排序
  - 合并排序
  - 堆排序
- 有更好的方法？
- 分治？

寻找顺序统计量问题

- 期望线性时间求解方法 $\Theta(n)$
- 分解：用到Random Partition
- 求解：递归处理
- 合并

寻找顺序统计量问题

● 期望线性时间求解方法 伪码

```
RAND-SELECT(A, p, q, i)  ▷ith smallest of A[p..q]
    if p = q then return A[p]
    r ← RAND-PARTITION(A, p, q)
    k ← r – p + 1  ▷k = rank(A[r])
    if i = k then return A[r]
    if i < k then return RAND-SELECT(A, p, r – 1, i)
    else return RAND-SELECT(A, r + 1, q, i – k)
```

寻找顺序统计量问题

- 期望线性时间求解方法 示例

Select the $i = 7$th smallest:

6	10	13	5	8	3	2	11

$i = 7$

pivot

Partition:

2	5	3	6	8	13	10	11

$k = 4$

Select the $7 - 4 = 3$rd smallest recursively.

寻找顺序统计量问题

- 期望线性时间求解方法 直观分析
- 一般情况 $T(n) = T(9n/10) + \Theta(n) = \Theta(n)$
- 最坏情况 $T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$

寻找顺序统计量问题

- 期望线性时间求解方法
- 可以证明，在平均情况下，任何顺序统计量可以在线性时间内得到

寻找顺序统计量问题

- 最坏情况线性时间
- 基本思想：保证对数组的划分是好的划分

寻找顺序统计量问题

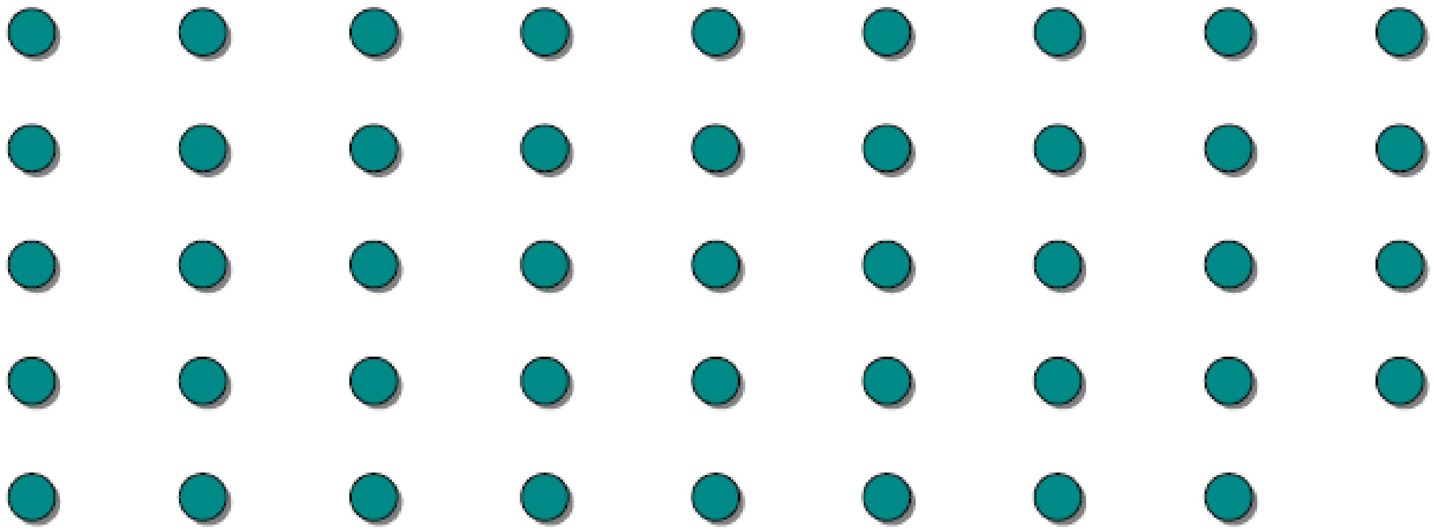
- SELECT的步骤
  - 1.将输入数组的n个元素分为$\lfloor n/5 \rfloor+1$组，其中$\lfloor n/5 \rfloor$组每组5个元素，余下的一组由剩下的n mod 5个元素组成；
  - 2.寻找这$\lfloor n/5 \rfloor+1$组中每一组的中位数（先对每组中的元素进行插入排序，然后从排序后的序列中选出中位数）；
  - 3.对第二步中找出的$\lfloor n/5 \rfloor+1$组中位数，递归调用SELECT以找到其中位数x；

寻找顺序统计量问题

- SELECT的步骤 续
  - 4.PARTITION，按中位数x对输入数组进行划分，x为第k小元素；
  - 5.如果i=k，则返回x；
  - 否则，如果i<k，则在低区递归调用SELECT寻找第i小元素
  - 否则，如果i>k，则 在高区寻找第(i-k)个最小元素

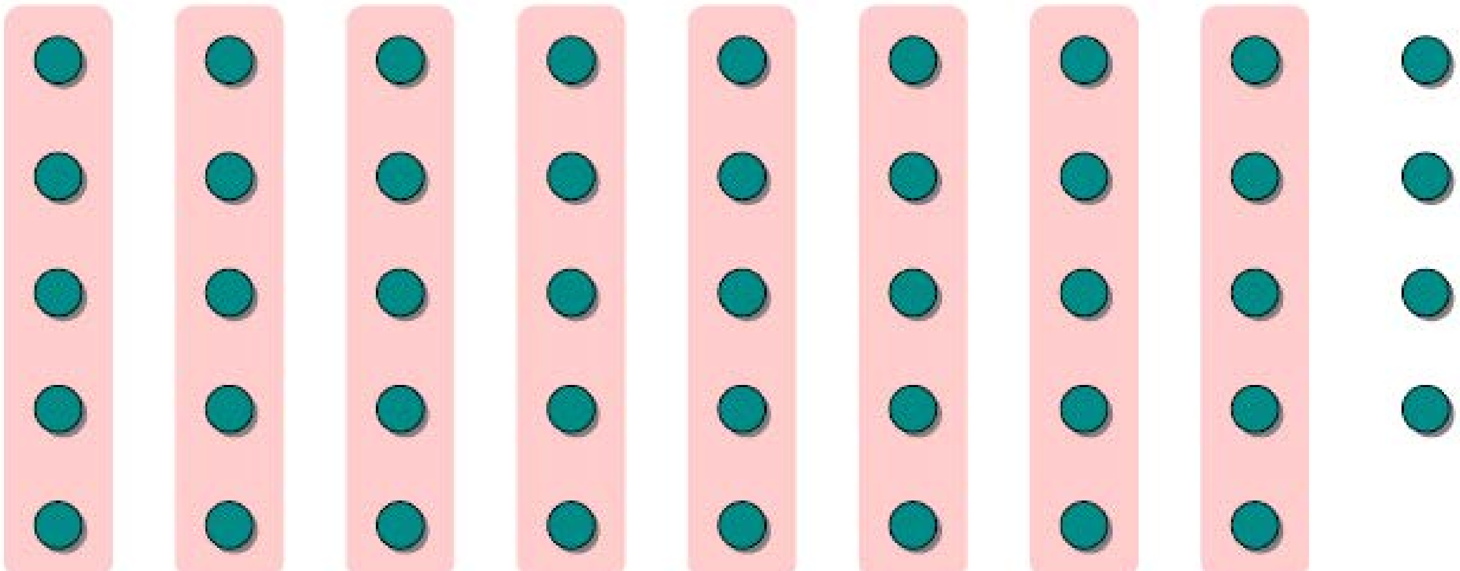
寻找顺序统计量问题

- 示例，n个元素



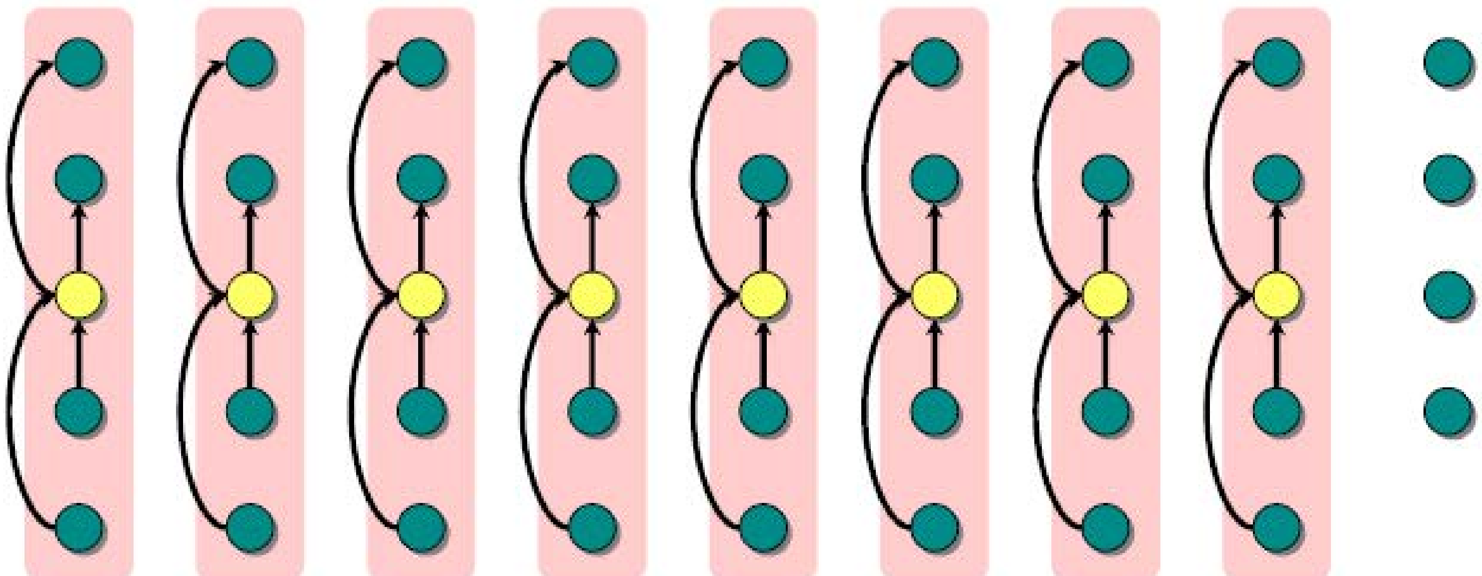
寻找顺序统计量问题

- 示例， n个元素分为$\lfloor n/5 \rfloor+1$组



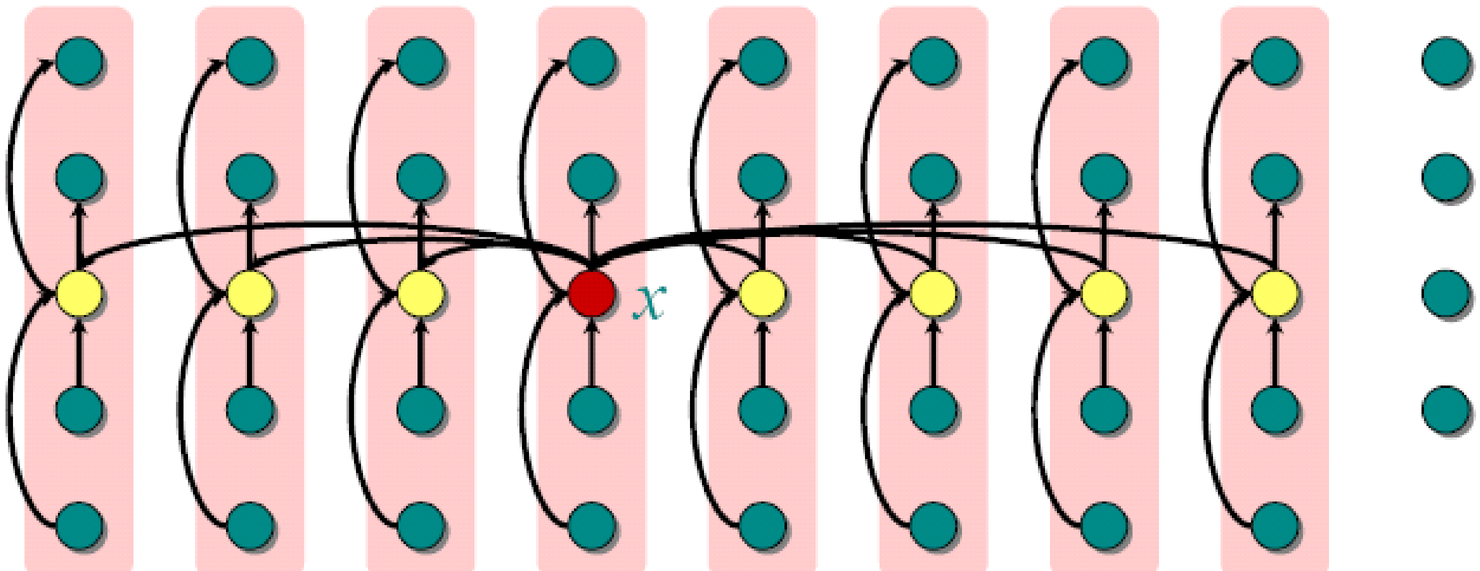
寻找顺序统计量问题

- 示例，寻找$\lfloor n/5 \rfloor$+1组中每一组的中位数



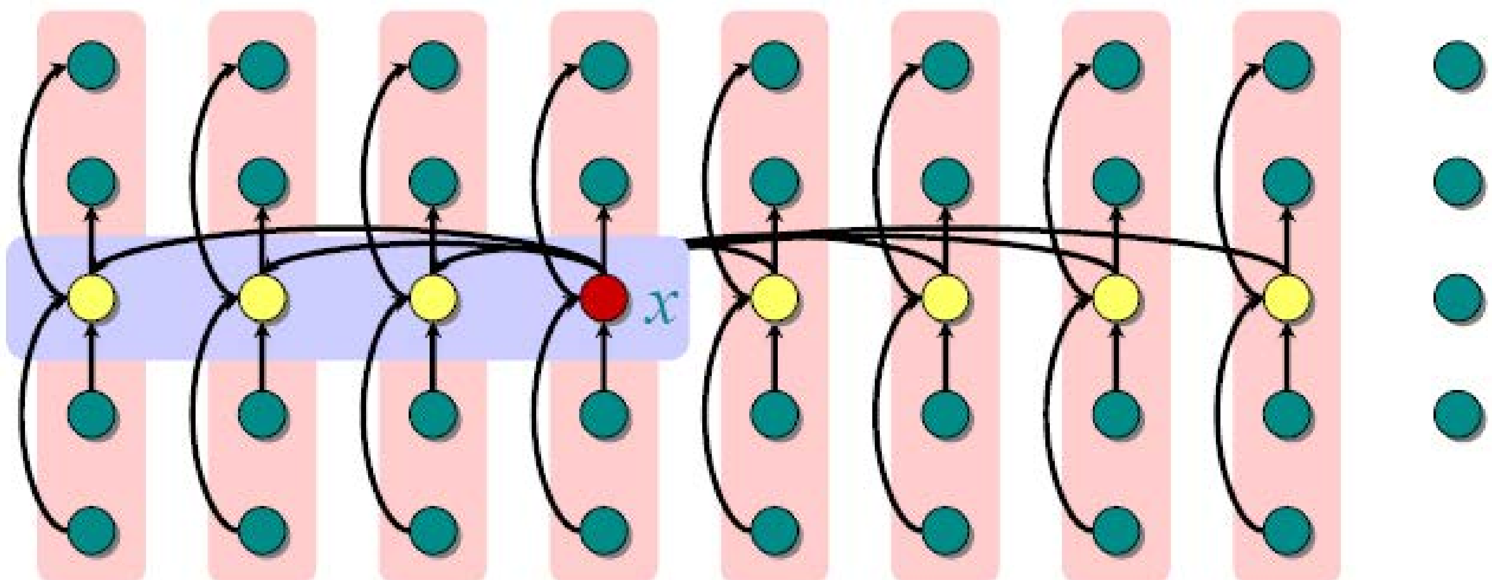
寻找顺序统计量问题

- 示例，递归调用SELECT找出$\lfloor n/5 \rfloor$的中位数x



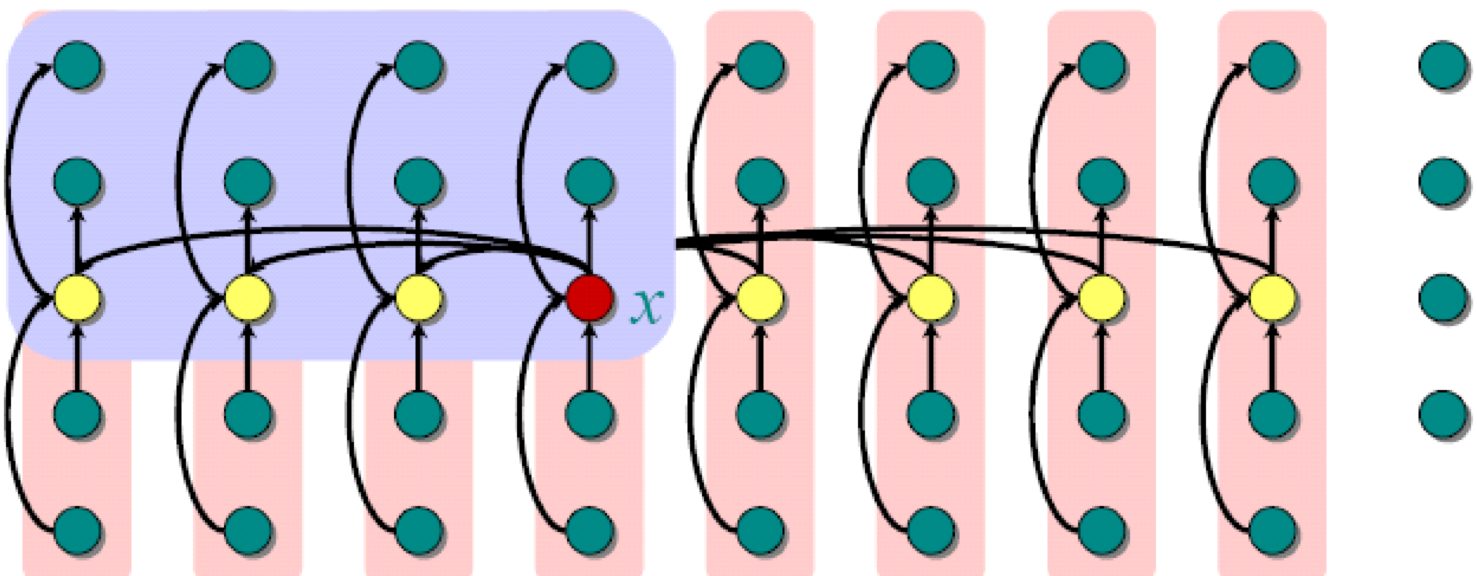
寻找顺序统计量问题

- 至少有一半组（$\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$组）的中位数小于或等于x



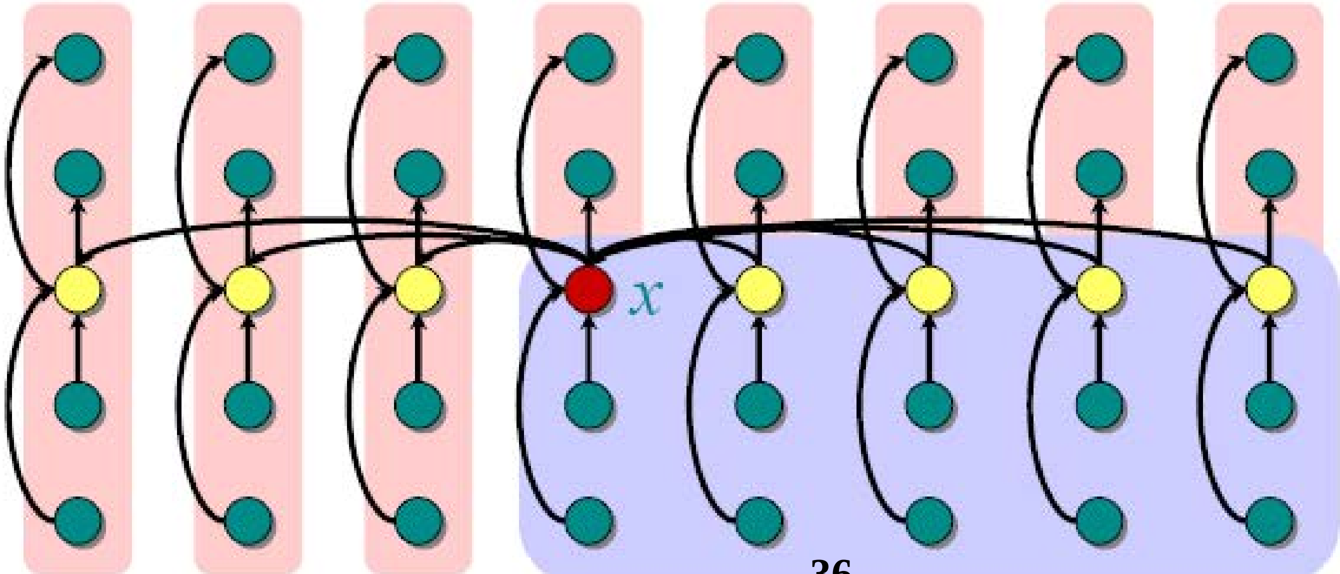
寻找顺序统计量问题

- 至少有一半组（$\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$组）的中位数小于或等于x
- 因此，至少$3\lfloor n/10 \rfloor$个元素$\leq$x



寻找顺序统计量问题

- 至少有一半组（$\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$组）的中位数小于或等于x
- 因此，至少3$\lfloor n/10 \rfloor$个元素≤x
- 至少3$\lfloor n/10 \rfloor$≥x



寻找顺序统计量问题

- 分析
  - 1.将输入数组的n个元素分为$\lfloor n/5 \rfloor+1$组；
  - 2.寻找这$\lfloor n/5 \rfloor+1$组中每一组的中位数；
  - 3.对第二步中找出的$\lfloor n/5 \rfloor+1$组中位数，递归调用SELECT以找到其中位数x；
  - 4.PARTITION，按中位数x对输入数组进行划分，x为第k小元素；
  - 5.如果i=k，则返回x；否则，如果i<k，则在低区递归调用SELECT寻找第i小元素；否则，如果i>k，则在高区寻找第(i-k)个最小元素。

分治法小结

- 二分搜索
- 幂乘
- 合并排序
- 快速排序
- Fibonacci数列
- Strassen矩阵乘法
- 最大元、最小元、寻找顺序统计量问题
- 最近点对问题
- ……