

1.

2300012929 尹锦河

2.7 设  $A$  是含有  $n$  个元素的数组, 如果元素  $x$  在  $A$  出现的次数大于  $n/2$ , 则称  $x$  是  $A$  的主元素.

(1) 如果  $A$  中元素是可以排序的, 设计一个  $O(n \log n)$  时间的算法, 判断  $A$  中是否存在主元素.

(2) 对于(1)中可排序的数组, 能否设计一个  $O(n)$  时间的算法?

(3) 如果  $A$  中元素只能进行“是否相等”的测试, 但是不能排序, 设计一个算法判断  $A$  中是否存在主元素.

对于问题(3)解法(前两问用此即可):

```
def FindPossible(A:[x1, ..., xn]):  
    if (n==1): return (x1, 1)  
    A1, A2 ← CutHalf(A)  
    (x1, c1), (x2, c2) ← FindPossible(A1), FindPossible(A2)  
    if x1 ≠ x2:  
        if c1 ≥ c2: return (x1, c1 - c2)  
        else: return (x2, c2 - c1)  
    else: return (x1, c1 + c2)
```

```
def calcCnt(A:[x1, ..., xn], x):  
    if (n==1): return (x1 == x)  
    A1, A2 ← CutHalf(A)  
    return calcCnt(A1, x) +  
           calcCnt(A2, x)
```

```
solve(A):  
    x, c ← FindPossible(A)  
    return calcCnt(A, x) > n/2
```

正确性: 如果有可能的主元素, 那么在其与其他元素一对一相消时, 一定会最后  
在留一个. 最后 check 即可.

时间:  $T(n) = 2T(n/2) + O(1) = O(n)$

2.27 如图 2.8 所示, 城市街道都是水平或垂直分布, 有  $m+1$  条, 不妨设任何两个相邻位置之间的距离都是 1. 在街道的十字路口有  $n$  个商店, 图中的  $n=3, m=8$ , 3 个商店的坐标位置(图中的圆点)分别是  $(2, 4), (5, 3), (6, 6)$ . 现在需要在某个路口位置建一个合用的仓库. 若仓库选择  $(3, 5)$  位置, 那么这 3 个商店到仓库的路程(只能沿街道行进)总长至少是 10. 设计一个算法找到仓库的最佳位置, 使得所有商店到仓库路程的总长达到最短.

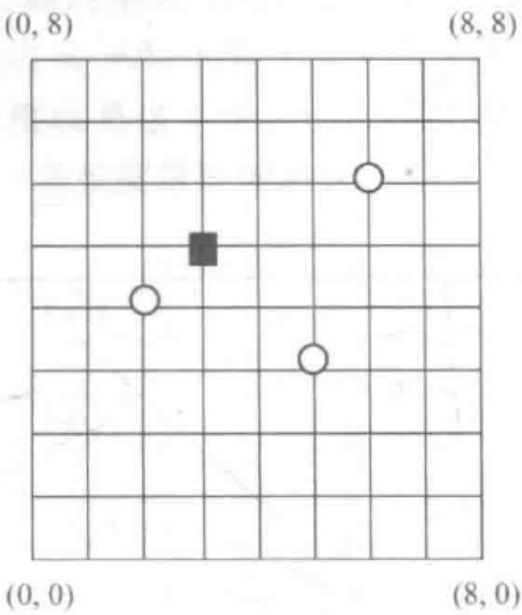


图 2.8 街道图

对于所有商店  $x$  坐标  $A[x_1, \dots, x_n]$ ,  
 $y$  坐标  $B[y_1, \dots, y_n]$ .

$A \leftarrow \text{sort}(A), B \leftarrow \text{sort}(B)$

取  $A, B$  中位数  $x', y'$  为仓库坐标即可

## 2 Forces Between Particles

You guys have learned about how to compute the electrostatic forces between two charged particles in high school physics. Consider a case where we have particles at points  $\{1, 2, \dots, n\}$  on the real axis, and the particle at point  $j$  has charge  $q_j$  (the charge can be either positive or negative). By Coulomb's law, the total force on each particle  $j$  is:

$$F_j = \sum_{i < j} \frac{C q_i q_j}{(j-i)^2} - \sum_{i > j} \frac{C q_i q_j}{(j-i)^2},$$

where  $C$  is a given constant (the Coulomb constant). Design an algorithm that computes all the forces  $F_j, j = 1, 2, \dots, n$  in  $O(n \log n)$  time.

令数组  $P \leftarrow [P_1, P_2, \dots, P_{2n-1}]$ , 其中  $P_i = \begin{cases} \frac{C}{(i-n)^2} & , i \neq n. \\ 0 & , i = n. \end{cases}$

$V \leftarrow \text{卷积}(P, Q)$ , [用FFT在  $O(n \log n)$  内完成]

$A = [V[i+n] \cdot Q[i] \cdot C, i=1, \dots, n]$  即为答案.

3.

(1)  $O(n^2)$ , 每次 pivot 恰好选到最大/最小元素.

(2) 在  $\lfloor \frac{n}{5} \rfloor$  组的中位数中, 有  $\lfloor \frac{n}{5} \rfloor / 2$  个元素  $> x$ .

相应组中有 2 个  $> x$  (中位数)  $\Rightarrow$  共  $\lfloor \frac{n}{5} \rfloor / 2 \times 3 = \frac{3n}{10} + 1 > x$

类似地, 有  $\frac{3n}{10} + 1 < x$

(3)  $T(n) = T(\frac{n}{5}) + T(\frac{7n}{10}) + O(n)$

不妨假设  $c > 3t$ ,  $T(x) \leq cn \quad \forall x < n$ .

$$T(n) = T(\frac{n}{5}) + T(\frac{7n}{10}) + tn$$

$$\leq \frac{9}{10} cn + tn$$

$$\leq cn$$

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

4. def skylines(lines):

$x \leftarrow$  斜率最大的 lines.

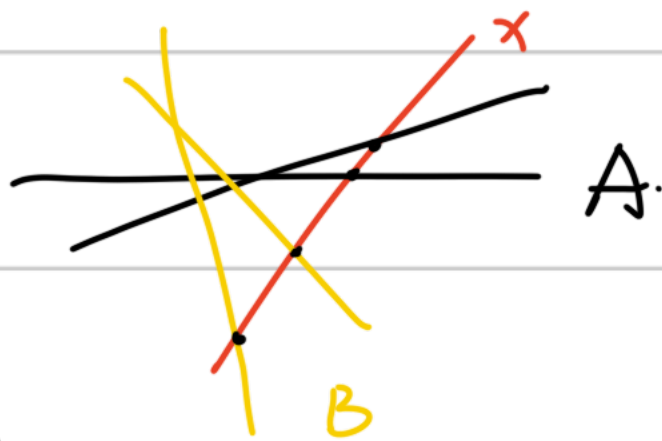
lines'  $\leftarrow$  lines/ $x$  且与  $x$  相交的线

A, B  $\leftarrow$  将 lines 按与  $x$  交点后中位  $x$  坐标切分  
(中位数查找  $O(n)$ ).

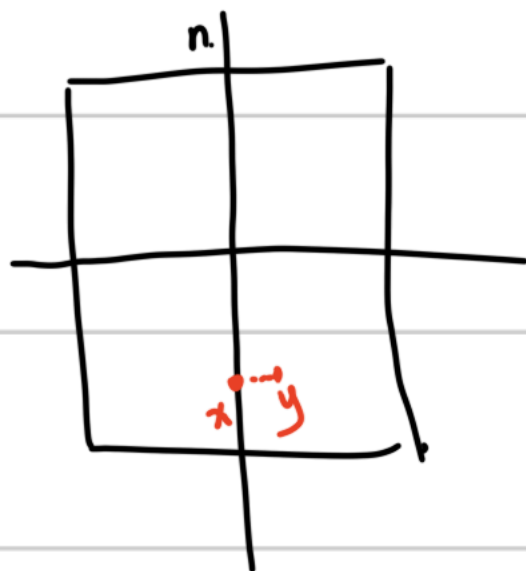
A', B'  $\leftarrow$  skylines(A'), skylines(B')

return {A', B', x}.

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) = O(n \log n)$$



5.



将中位线(横竖)元素遍历. 找到最小的  $x$   
且  $x$  周围最小为  $y$  (若无, 则结束, 找到)  
问题缩小到  $y$  对应的块  $(\frac{n}{2} \times \frac{n}{2})$

$$T(n) = T\left(\frac{n}{2}\right) + O(2n) = O(n)$$