

算法设计与分析



蒋婷婷

元素选择问题

问题：从给定的集合 L 中选择第 i 小的元素
不妨设 L 为 n 个不等的实数

$i=1$, 称为最小元素;

$i=n$, 称为最大元素;

位置处在中间的元素, 称为中位元素

当 n 为奇数时, 中位数只有1个, $i=(n+1)/2$;

当 n 为偶数时, 中位数有2个, $i=n/2, n/2+1$.

选最大

输入： n 个不等的数

输出： max

算法4 Findmax

1. $max \leftarrow L[1];$
2. for $i \leftarrow 2$ to $length[L]$
3. do if $max < L[i]$
4. then $max \leftarrow L[i]$
5. return max

算法最坏情况下的时间复杂性为 $O(n)$

找最大和最小

通常算法：顺序比较

复杂性： $W(n)=2n-3$

算法5 FindMaxMin

1. 将 n 个元素两两一组分成 $\lfloor n/2 \rfloor$ 组
2. 每组比较，得到 $\lfloor n/2 \rfloor$ 个较小和 $\lfloor n/2 \rfloor$ 个较大
3. 在 $\lceil n/2 \rceil$ 个(n 为奇数，是 $\lfloor n/2 \rfloor + 1$)较小中找最小 min
4. 在 $\lceil n/2 \rceil$ 个(n 为奇数，是 $\lfloor n/2 \rfloor + 1$)较大中找最大 max

复杂性：行2 比较 $\lfloor n/2 \rfloor$ 次，行3--4 比较至多 $2\lceil n/2 \rceil - 2$ 次，

$$W(n) = \lfloor n/2 \rfloor + 2\lceil n/2 \rceil - 2 = n + \lceil n/2 \rceil - 2 = \lceil 3n/2 \rceil - 2$$

找第二大

通常算法：顺序比较

1. 顺序比较找到最大 max ;
2. 从剩下的 $n-1$ 个数中找最大，就是第二大 $second$

复杂性： $W(n)=n-1+n-2=2n-3$

锦标赛算法：

两两分组比较，大者进入下一轮

每个元素用数表记录每次比较时小于自己的元素

锦标赛算法

1. $k \leftarrow n$
2. 将 k 个元素两两一组，分成 $\lfloor k/2 \rfloor$ 组
3. 每组的2个数比较，找到较大的数
4. 将被淘汰的较小的数在淘汰它的数所指向的链表中做记录
5. if k 为奇数 then $k \leftarrow \lfloor k/2 \rfloor + 1$
6. else $k \leftarrow \lfloor k/2 \rfloor$
7. if $k > 1$ then goto 2
8. $max \leftarrow$ 最大数
9. $second \leftarrow max$ 的链表中的最大

复杂性:

$$W(n) = n - 1 + \lceil \log n \rceil - 1 = n + \lceil \log n \rceil - 2$$

一般性选择问题

问题描述

输入：数组 L , L 的长度 n , 正整数 k , $1 \leq k \leq n$.

输出：第 k 小的数

通常算法

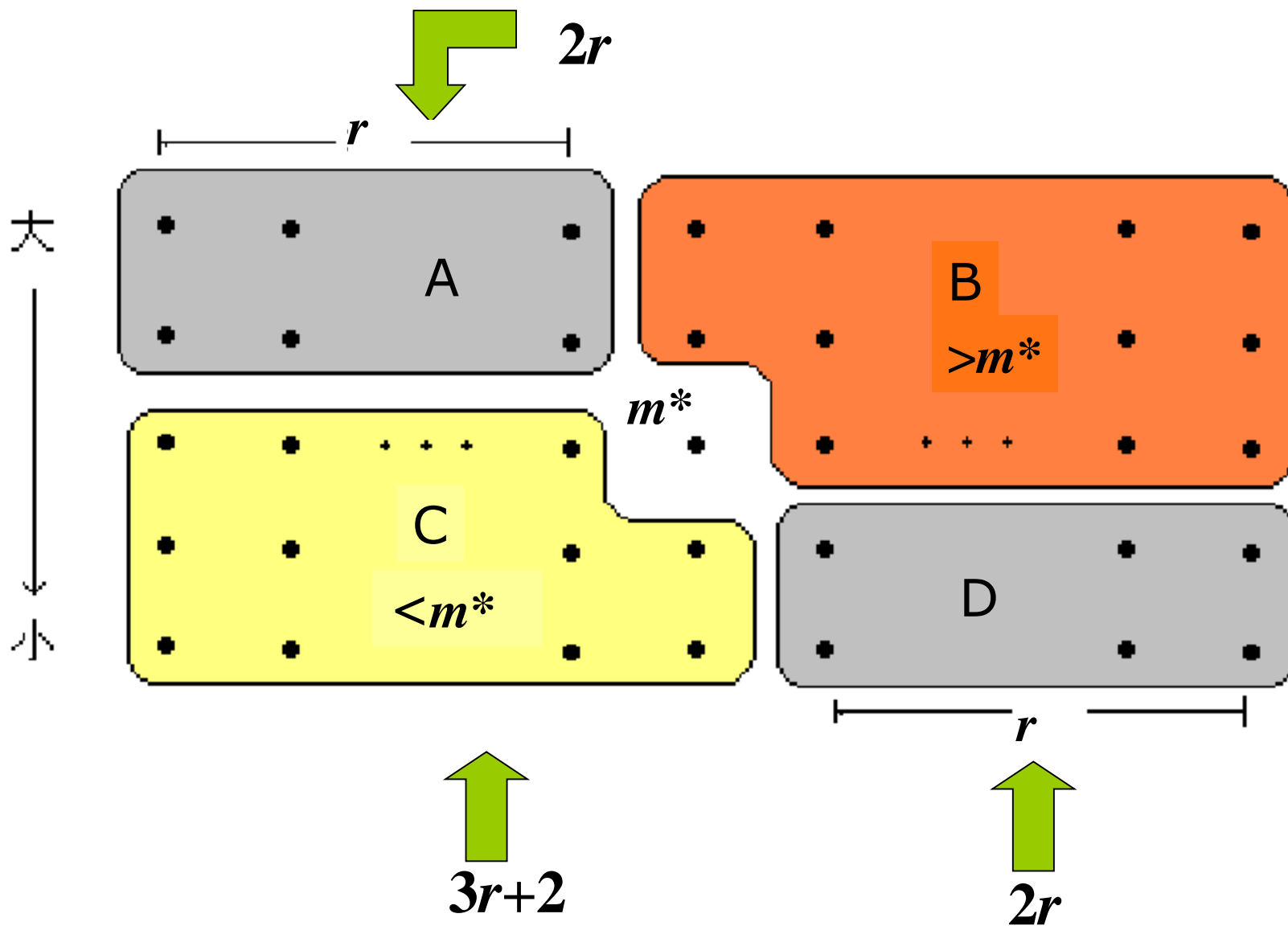
1. 排序
2. 找第 k 小的数

时间复杂性： $O(n \log n)$

分治选择算法

算法 $\text{Select}(S, k)$

1. 将 S 划分成5个一组, 共 $n_M = \lceil n/5 \rceil$ 个组
2. 每组找中位数, n_M 个中位数放到集合 M .
3. $m^* \leftarrow \text{Select}(M, \lceil |M|/2 \rceil)$ 将 S 中的数划分成 A, B, C, D 四个集合
4. 把 A 和 D 中的每个元素与 m^* 比较, 小的构成 S_1 , 大的构成 S_2 ;
5. $S_1 \leftarrow S_1 \cup C$; $S_2 \leftarrow S_2 \cup B$;
6. if $k = |S_1| + 1$ then 输出 m^*
7. else if $k \leq |S_1|$
8. then $\text{Select}(S_1, k)$
9. else $\text{Select}(S_2, k - |S_1| - 1)$



最坏情况：子问题大小为 $2r + 2r + 3r + 2 = 7r + 2$

复杂度估计 $W(n)=O(n)$

不妨设 $n=5(2r+1)$, $|A|=|D|=2r$, $r = \frac{\frac{n}{5}-1}{2} = \frac{n}{10} - \frac{1}{2}$

算法工作量

行2: $O(n)$

行3: $W(n/5)$

行4: $O(n)$

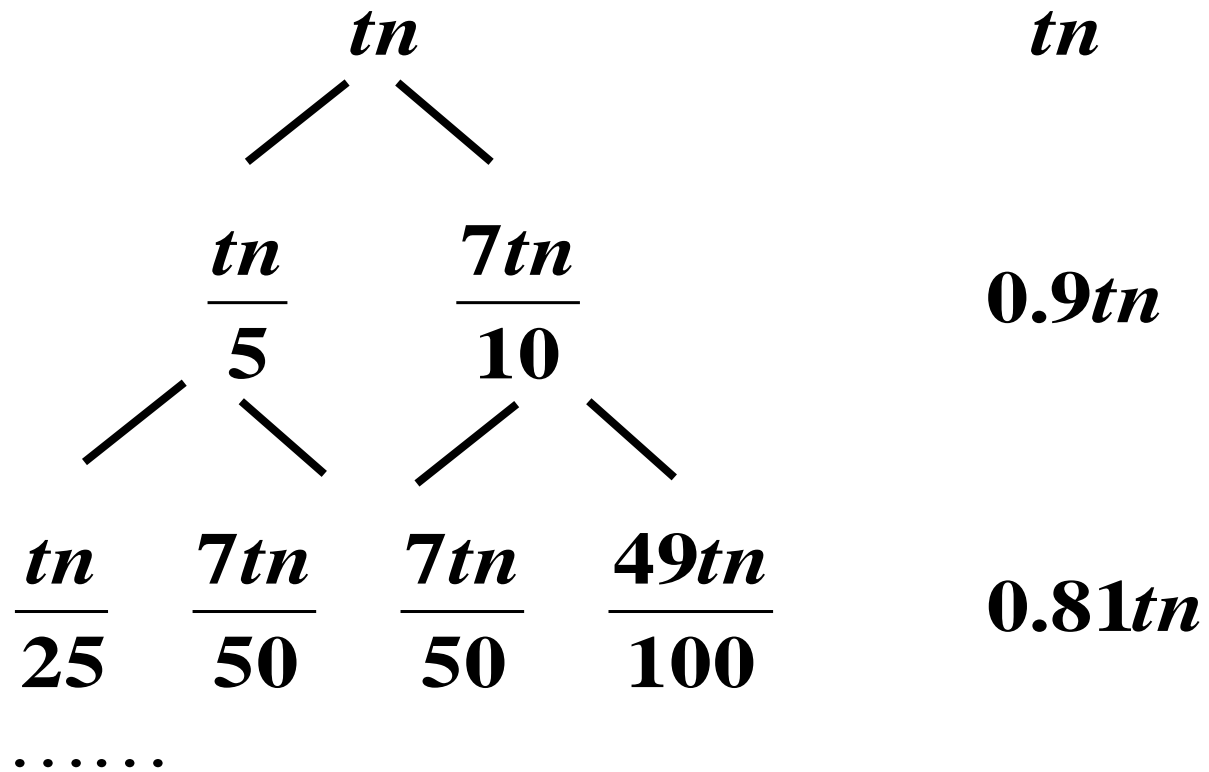
行8-9: $W(7r+2)$

$$\begin{aligned} W(7r+2) &= W\left(7\left(\frac{n}{10} - \frac{1}{2}\right) + 2\right) \\ &= W\left(\frac{7n}{10} - \frac{3}{2}\right) \leq W\left(\frac{7n}{10}\right) \end{aligned}$$

用递归树做复杂度估计

$$W(n) \leq W\left(\frac{n}{5}\right) + W\left(\frac{7n}{10}\right) + cn \leq cn + \frac{9}{10}cn + \frac{81}{100}cn + \dots = O(n)$$

递归树



分治策略应用：卷积

□ 向量计算：

给定向量 $a = (a_0, a_1, \dots, a_{n-1})$ 和 $b = (b_0, b_1, \dots, b_{n-1})$

向量和 $a + b = (a_0 + b_0, a_1 + b_1, \dots, a_{n-1} + b_{n-1})$

内积 $a \cdot b = a_0 b_0 + a_1 b_1 + \dots + a_{n-1} b_{n-1}$

卷积 $a * b = (c_0, c_1, \dots, c_{2n-2})$, 其中

$$c_k = \sum_{\substack{i+j=k \\ i,j < n}} a_i b_j, \quad k = 0, 1, \dots, 2n-2$$

$a_0 b_0$	$a_0 b_1$	$a_0 b_{n-2}$	$a_0 b_{n-1}$
$a_1 b_0$	$a_1 b_1$	$a_1 b_{n-2}$	$a_1 b_{n-1}$
.....
$a_{n-1} b_0$	$a_{n-1} b_1$	$a_{n-1} b_{n-2}$	$a_{n-1} b_{n-1}$

卷积等价于多项式相乘

多项式乘法:

给定多项式

$$A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1}$$

$$B(x) = b_0 + b_1x + b_2x^2 + \dots + b_{n-1}x^{n-1}$$

计算 $C(x) = A(x) B(x)$

$$\begin{aligned} &= a_0b_0 + (a_0b_1 + a_1b_0)x + (a_0b_2 + a_1b_1 + a_2b_0)x^2 \\ &\quad + \dots + a_{m-1}b_{n-1}x^{m+n-2} \end{aligned}$$

其中 x^k 的系数

$$c_k = \sum_{\substack{i+j=k \\ i \in \{0,1,\dots,m-1\} \\ j \in \{0,1,\dots,n-1\}}} a_i b_j, \quad k = 0, 1, \dots, m+n-2$$

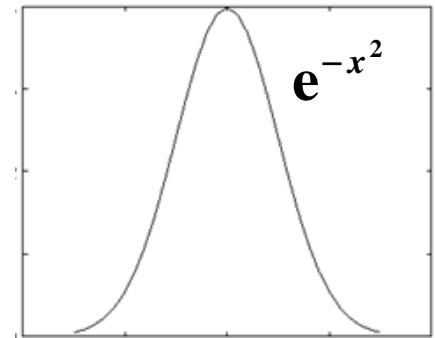
卷积应用：信号处理

给定序列 $a=(a_0,a_1,\dots,a_{m-1})$ ，表示信号在时刻 $0,1,\dots,m-1$ 的度量值，由于噪音干扰，需要平滑处理，即对值 a_i ，用其前后 k 步内的值进行加权平均。离 a_i 越近权值越大，处理结果为 a_i' 。

高斯滤波的权值函数为

$$w_s = \frac{1}{z} e^{-s^2}, \quad s = 0, \pm 1, \dots, \pm k$$

$$w = (w_{-k}, \dots, w_{-1}, w_0, w_1, \dots, w_k)$$



其中 z 用于归一化处理，使得所有的权值之和为1。

处理结果为

$$a_i' = \sum_{s=-k}^k a_{i+s} w_s$$

实例

$$a=(a_0,a_1,\dots,a_8), k=2, w=(w_{-2},w_{-1},w_0,w_1,w_2)=(b_4,b_3,b_2,b_1,b_0)$$

$$a_i'=a_{i-2}b_4+a_{i-1}b_3+a_ib_2+a_{i+1}b_1+a_{i+2}b_0, \text{ 下标之和为 } i+k$$

a_0b_0	a_0b_1	a_0b_2	a_0b_3	a_0b_4	a_2'
a_1b_0	a_1b_1	a_1b_2	a_1b_3	a_1b_4	a_3'
a_2b_0	a_2b_1	a_2b_2	a_2b_3	a_2b_4	a_4'
a_3b_0	a_3b_1	a_3b_2	a_3b_3	a_3b_4	a_5'
a_4b_0	a_4b_1	a_4b_2	a_4b_3	a_4b_4	a_6'
a_5b_0	a_5b_1	a_5b_2	a_5b_3	a_5b_4	
a_6b_0	a_6b_1	a_6b_2	a_6b_3	a_6b_4	
a_7b_0	a_7b_1	a_7b_2	a_7b_3	a_7b_4	
a_8b_0	a_8b_1	a_8b_2	a_8b_3	a_8b_4	

平滑处理

定义向量

$$a=(a_0, a_1, \dots, a_{m-1})$$

$$b=(b_{2k}, b_{2k-1}, \dots, b_0)$$

$$=(w_{-k}, w_{-k+1}, \dots, w_k)=w$$

$$a_i' = \sum_{s=-k}^k a_{i+s} b_{k-s} = \sum_{s=-k}^k a_{i+s} w_s$$

把权向量 $b=w$ 看作 $2k+1$ 长度的窗口
在 a 上移动计算卷积.

卷积计算

给定向量 $a = (a_0, a_1, \dots, a_{n-1})$ 和 $b = (b_0, b_1, \dots, b_{n-1})$

直接计算: $O(n^2)$

快速算法: 卷积计算等价于多项式相乘.

$$A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1}$$

$$B(x) = b_0 + b_1x + b_2x^2 + \dots + b_{n-1}x^{n-1}$$

$$C(x) = A(x)B(x) = a_0b_0 + (a_0b_1 + a_1b_0)x + (a_0b_2 + a_1b_1 + a_2b_0)x^2 \\ + (\dots + a_{m-1}b_{n-1})x^{m+n-2}, \quad C(x) \text{ 的系数向量就是 } a * b.$$

设计思想:

1. 选择值 x_1, x_2, \dots, x_{2n} , 求出 $A(x_j)$ 和 $B(x_j)$, $j=1, 2, \dots, 2n$
2. 对每个 j , 计算 $C(x_j)$
3. 利用多项式插值方法, 根据 $C(x)$ 在 $x=x_1, x_2, \dots, x_{2n}$ 的值求出多项式 $C(x)$

$2n$ 个数的选择: 1 的 $2n$ 次根

$$\omega_j = e^{\frac{2\pi j}{2n}i} = e^{\frac{\pi j}{n}i} = \cos \frac{\pi j}{n} + i \sin \frac{\pi j}{n} \quad j=0,1,\dots,2n-1$$

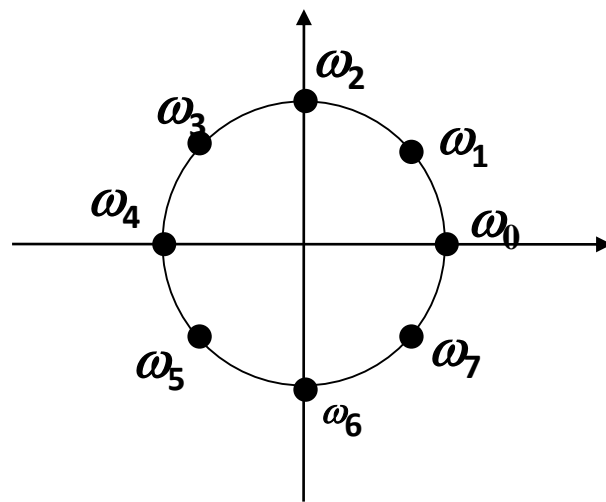
例如 $n=4$, 1的8次方根是:

$$\omega_0 = 1, \quad \omega_1 = e^{\frac{\pi i}{4}} = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i,$$

$$\omega_2 = e^{\frac{2\pi i}{4}} = i, \quad \omega_3 = e^{\frac{3\pi i}{4}} = -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i,$$

$$\omega_4 = e^{\pi i} = -1, \quad \omega_5 = e^{\frac{5\pi i}{4}} = -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i,$$

$$\omega_6 = e^{\frac{6\pi i}{4}} = -i, \quad \omega_7 = e^{\frac{7\pi i}{4}} = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i$$



多项式求值算法

给定多项式: $A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$

设 x 为1的 $2n$ 次方根, 对所有的 x 计算 $A(x)$ 的值.

算法1: 对每个 x 做下述运算:

依次计算每个项 $a_i x^i$, 对 i 求和得到 $A(x)$,

$$T_1(n) = O(n^3)$$

算法2: $A_1(x) = a_{n-1}$

$$A_2(x) = a_{n-2} + xA_1(x)$$

$$A_3(x) = a_{n-3} + xA_2(x)$$

...

$$A_n(x) = a_0 + xA_{n-1}(x) = A(x)$$

$$T_2(n) = O(n^2)$$

分治法：多项式求值

原理：

$$A_{\text{even}}(x) = a_0 + a_2x + a_4x^2 + \dots + a_{n-2}x^{(n-2)/2}$$

$$A_{\text{odd}}(x) = a_1 + a_3x + a_5x^2 + \dots + a_{n-1}x^{(n-2)/2}$$

$$A(x) = A_{\text{even}}(x^2) + x A_{\text{odd}}(x^2), \quad x^2 \text{ 为 } 1 \text{ 的 } n \text{ 次根}$$

算法3：

1. 计算1 的所有的 $2n$ 次根
2. 分别计算 $A_{\text{even}}(x^2)$ 与 $A_{\text{odd}}(x^2)$
3. 利用步2 的结果计算 $A(x)$

复杂度分析： $T(n) = T_1(n) + f(n)$, $f(n) = O(n)$ 计算 $2n$ 次根时间

$$T_1(n) = 2T_1(n/2) + g(n), \quad g(n) = O(n),$$

$$T_1(1) = O(1)$$

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

卷积计算的时间分析

步1: 求值 $A(\omega_j)$ 和 $B(\omega_j)$, $j=0,1,\dots,2n-1$. $O(n\log n)$

步2: 计算 $C(\omega_j)$, $j=0,1,\dots,2n-1$. $O(n)$

步3: 构造多项式

$$D(x)=C(\omega_0)+C(\omega_1)x+\dots+C(\omega_{2n-1})x^{2n-1}$$

可以证明: $D(\omega_0)=2nc_0$

$$D(\omega_j)=2nc_{2n-j}, \quad j=1, \dots, 2n-1$$

即只要计算出所有的 $D(\omega_j)$, 就得到所有的 c_{2n-j} ,
 $j=0,1,\dots,2n-1$. $O(n\log n)$

总的时间为 $O(n\log n)$

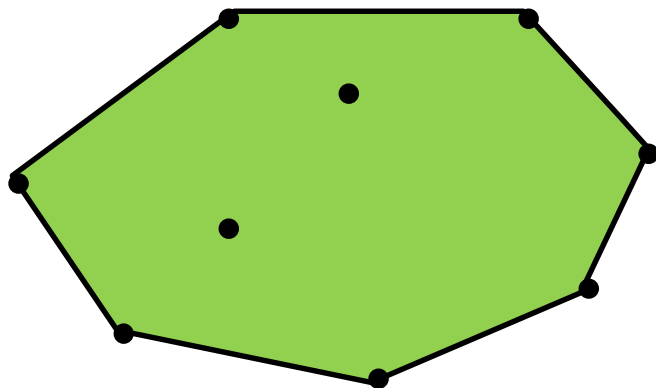
平面点集的凸包

- 背景

图形处理中用于形状识别：字形识别、碰撞检测
平面凸包算法是核心算法

- 问题（平面凸包）

给定大量离散点的集合 Q ，求一个最小的凸多边形，使得 Q 中的点在该多边形内或者边上.



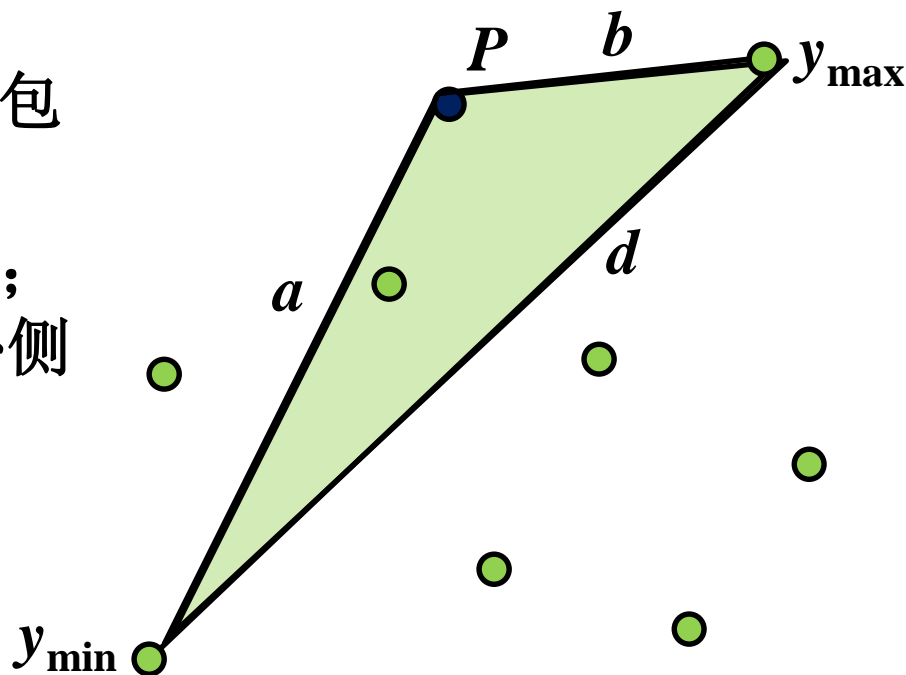
分治算法

算法设计思想

1. 以 $d=\{y_{\max}, y_{\min}\}$ 划分 Q 为左点集 L 和右点集 R
2. **Deal**(L); **Deal**(R)

Deal(L)

1. 把距离 d 最远点 P 加入凸包
2. 检查 L 中其他点是否在三角形内；在则从 L 中删除；否则根据在 a 或 b 边的外侧划分在两个子问题中
3. **Deal**(a)
4. **Deal**(b)



算法分析

- 初始用 d 划分 $O(n)$
- Deal 递归调用
 - 找凸包顶点 P $O(n)$
 - 确定点是否在三角形内部 $O(n)$

$$W(n) = W(n-1) + O(n)$$

- 最坏情况为 $O(n^2)$

- $T(n) = O(n) + W(n) = O(n^2)$