改进分治算法的途径2：增加预处理

例子：平面点对问题

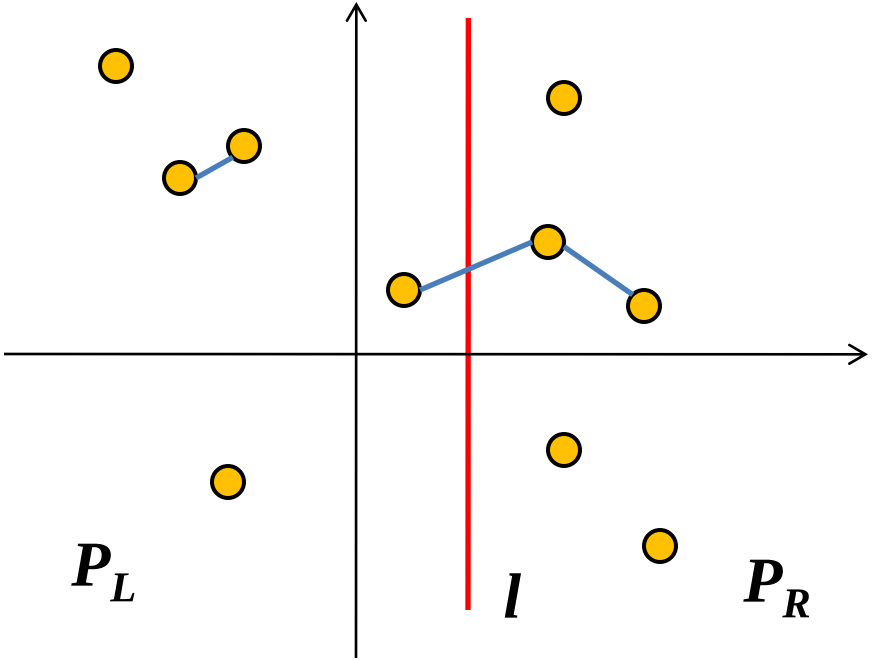
输入: 平面点集 P 中有n 个点, $n>1$
输出: P中的两个点，其距离最小

蛮力算法:
$C(n,2)$个点对,计算最小距离，$O(n^2)$

分治策略: P 划为大小相等的P_L和P_R

1. 分别计算 P_L、P_R中最近点对
2. 计算 P_L与P_R中各一个点的最近点对
3. 上述情况下的最近点对是解

划分实例：n=10



算法伪码

MinDistance (P, X, Y)

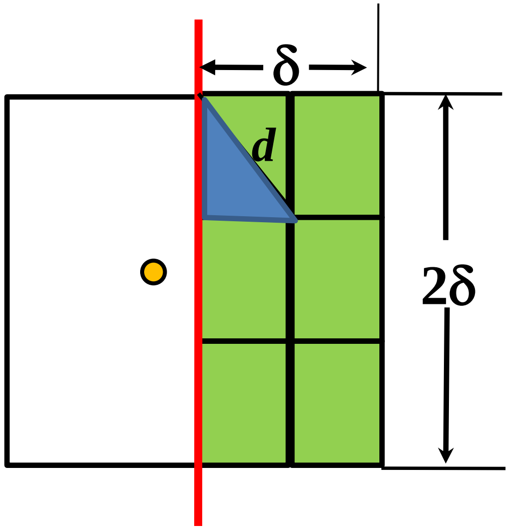
输入: 点集P, X和Y为横、纵坐标数组

输出: 最近的两个点及距离

1. 若$|P|\le 3$，直接计算其最小距离
2. 排序X,Y
3. 做中垂线 l 将P划分为P_L和P_R
4. **MinDidtance** (P_L,X_L,Y_L)
5. **MinDistance** (P_R,X_R,Y_R)
6. $\delta=\min(\delta_L,\delta_R)$//$\delta_L,\delta_R$为子问题的距离
7. 检查距 l 不超过δ两侧各1个点的距离．若小于δ，修改 δ为这个值

跨边界处理

$$d=\sqrt{(\delta/2)^2+(2\delta/3)^2}$$
$$=\sqrt{\delta^2/4+4\delta^2/9}$$
$$=\sqrt{25\delta^2/36}=5\delta/6$$



右边每个小方格至多1个点，每个点至多比较对面的6个点，检查1个点是常数时间，$O(n)$ 个点需要$O(n)$时间

算法分析

步1 递归边界处理：　$O(1)$

步2 排序：　$O(n\log n)$

步3 划分：　$O(1)$

步4-5子问题：　$2T(n/2)$

步6确定δ:　$O(1)$

步7检查跨边界点对：　$O(n)$

$$\boxed{\begin{array}{l} T(n)=2T(n/2)+O(n\log n) \\ T(n)=O(1),\ n\leq 3 \end{array}}$$

递归树求解　$T(n)=O(n\log^2 n)$

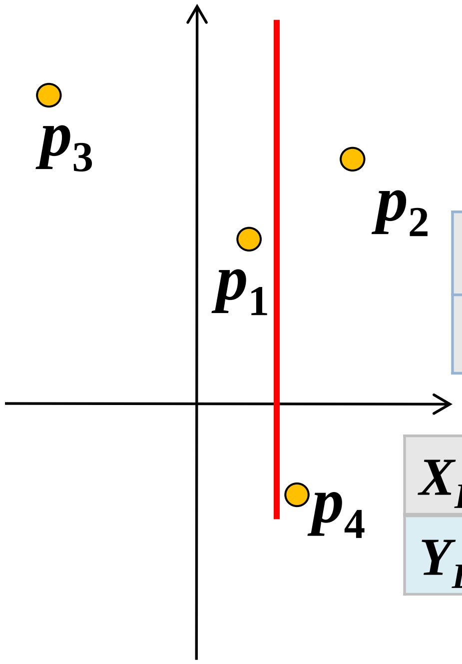
增加预处理

原算法：
在每次划分时对子问题数组重新排序

改进算法：
1. 在递归前对 X,Y 排序，作为预处理
2. 划分时对排序的数组 X,Y 进行拆分，
   得到针对子问题 P_L 的数组 X_L,Y_L 及
   针对子问题 P_R 的数组 X_R,Y_R

原问题规模为 n，拆分的时间为 $O(n)$

实例：递归中的拆分



P	1	2	3	4
X	0.5	2	-2	1
Y	2	3	4	-1

输入

X	-2(3)	0.5(1)	1(4)	2(2)
Y	-1(4)	2(1)	3(2)	4(3)

预处理

X_L	2(3)	0.5(1)
Y_L	2(1)	4(3)

X_R	1(4)	2(2)
Y_R	-1(4)	3(2)

拆分后

改进算法时间复杂度

$W(n)$为算法时间复杂度
递归过程：$T(n)$，预处理：$O(nlogn)$

$$W(n) = T(n) + O(nlogn)$$
$$T(n) = 2T(n/2) + O(n)$$
$$T(n) = O(1) \qquad n \le 3$$

解得 $T(n) = O(nlogn)$

于是 $W(n) = O(nlogn)$

小结

- 依据

$$W(n) = aW(n/b) + f(n)$$

- 提高算法效率的方法：
  - 减少子问题个数a：

$$W(n)=O(n^{\log_b a})$$

  - 增加预处理，减少$f(n)$