第二章　基本策略

2.1　分治

Problem 2.1.1
有n个项组成的数组$R[1\cdots n]$，唯一可以对R进行的操作是$Check(R[i], R[j])$，返回True，如果R[i]和R[j]相等；返回False，如果R[i] 和R[j]不等。对于一个有n个项的数组R，如果至少有$\frac{n}{13}$项同项e相等，那么e被称为常见项，我们希望设计一个使用$O(n\log n)$ 个Check操作调用的算法返回所有常见项。

(a) 请设计一个分治算法找到所有的常见项

  (i) 证明R中存在最多13个不同的常见项。

  (ii) 设e是$R[1\cdots n]$中的常见项，证明e至少是$R[1\cdots \frac{n}{2}]$和$R[\frac{n}{2}+1\cdots n]$两个数组中一个数组的常见项。

  (iii) 请根据上面的讨论设计一个分支算法。

(b) 常见项定义中的13换为任意大于等于2的正整数k，算法是否还正常工作？

(c) 当上述常数k被设定为常数2时（即要求找到出现次数至少有$\frac{n}{2}$的项），是否有更高效的算法解决该问题？

(d) （选做）常见项问题的难度（下界）是什么？它是否比比较排序更难或更容易？

Problem 2.1.2 (寻找maxima)
给定二维平面上的n个点$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$。我们定义“$(x_1, y_1)$支配$(x_2, y_2)$”，如果$x_1 > x_2, y_1 > y_2$。一个点被称作为maxima，如果没有任何其它点支配它。例如图2.1 中画圈的点均是maxima。

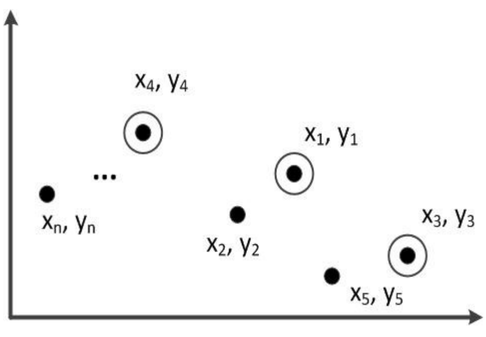

图 2.1: 平面上的maxima点

(a) 请设计一个算法找出所有的maxima。

- 你可以对点进行某种排序，进而得到一个简便的算法。
- 如果不允许对点进行排序，你能否设计一个分治算法？

(b) 有人声称他用如下的算法可以在$O(n)$的时间内解决该问题。他的思路是取x坐标和y坐标的中值点，将所有点分成四个象限，然后有两种不同的思路均可以得到$O(n)$的算法：

- 对每个象限递归地解决问题，再将所有象限的结果综合。依据这一思路，算法的代价应该满足$T(n) = 3T(\frac{1}{4}n) + O(n)$，由master定理知$T(n) = O(n)$。
- 易知左下象限的点一定不可能是maxima，所以每次划分都可以去掉左下象限的$\frac{1}{4}$的点。因此算法的代价应该满足$T(n) = T(\frac{3}{4}n) + O(n)$，由master定理知$T(n) = O(n)$。

请将上述思路整理成一个算法，并证明其正确性；或者找出上述思路的错误。

(c) （选做）如果你能找出上述思路的错误，那你能否进一步证明该问题不可能有$o(nlogn)$的算法（只能基于坐标的比较来确定一个点是否是maxima）。

Problem 2.1.3 (整数乘法)
给定两个整数，我们要求它们的乘积。这里我们考虑的是任意长度的整数，它的长度远大于机器的字长，因而我们要有专门的数据结构（例如数组）来存储它，并且我们度量相乘算法代价的关键操作为比特操作的个数。算法的输入输出分别为：

输入：	两个n-bit的整数x和y
输出：	乘积$x \cdot y$

请给出两个分别符合下面要求的算法来计算两个整数的乘积：

(a) $O(n^2)$。

(b) $o(n^2)$。

Problem 2.1.4 (矩阵乘法)
给定两个大小为$n \times n$的矩阵，求它们的乘积。假设矩阵中均为整数，两个整数的相乘/相加为关键操作。

(a) $O(n^3)$。

(b) $o(n^3)$。

Problem 2.1.5 (距离最近的点对)
给定平面上的n个点，请找出距离最近的一对点。

Problem 2.1.6
*Diogenes*教授有n个被认为是完全相同的VLSI芯片，原则上它们是互相测试的。教授的测试装置一次可测二片，当该装置中放有两片芯片时，每一片就对另一片测试并报告其好坏。一个好的芯片总能够报告另一片的好坏，但一个坏的芯片的结果是不可靠的。这样，每次测试的四种可能结果如下：

A芯片报告	B芯片报告	结论
B是好的	A是好的	都是好的，或都是坏的
B是好的	A是坏的	至少一片是坏的
B是坏的	A是好的	至少一片是坏的
B是坏的	A是坏的	至少一片是坏的

(a) 证明若多于$n/2$个芯片是坏的，在这种成对测试方式下，使用任何策略都不能确定哪个芯片是好的。假设坏的芯片可以联合起来欺骗教授。

(b) 假设有多于$n/2$个芯片是好的，考虑从n片中找出一片好芯片的问题。证明$\lfloor n/2 \rfloor$对测试就足以使问题的规模降至近原来的一半。

(c) 假设有多于$n/2$个芯片是好的，证明好的芯片可用$\Theta(n)$对测试找出。给出并解答表达测试次数的递归式。