第三章 习题

3.1 给定一个长度为奇数的有序整数数组，其中除一个元素之外，其他元素均恰好出现两次。试设计一个分治算法以 $o(n)$时间找出仅出现一次的元素。用自然语言表达算法思想，然后给出算法描述，并分析算法复杂度。

3.2 给定整数数组 A[0:n-1], A[i]∈Z。试设计一个分治算法，输出 A 中的最大连续子序列之和，亦即找出 0≤i,j≤n-1 使得 A[i]+A[i+1]+…+A[j]达到最大。用自然语言表达算法思想，然后给出算法描述，并分析算法复杂度。

3.3 给定一个由 n 个实数构成的集合 S 和另一个实数 x，判断 S 中是否有两个元素的和为 x。用自然语言表达算法思想，然后给出算法描述，并分析算法复杂度。

3.4 给定非负数组 $A[1:n]$，$B[1:m]$ 和整数 k。试设计一个算法计算集合 $\{A[i]\cdot B[j] \mid 1\le i\le n, 1\le j\le m\}$中的第 k 小的元素。用自然语言表达算法思想，然后给出算法描述，并分析算法复杂度。

附加思考题(供初学者思考练习，不用上交)

3.1 根据表达式 $a^{2k}=a^k\cdot a^k$ 和 $a^{2k+1}=a^k\cdot a^k\cdot a$ 设计分治(或递归)算法求解下列问题，并分析算法的时间复杂度。

(a)输入实数 a 和自然数 n,输出实数 a^n;

(b)输入实数矩阵 A 和自然数，输出实数矩阵 A^n.

3.2 斐波那契数列满足递归方程 $F(n+2)=F(n+1)+F(n)$, 其中 $F(0)=F(1)=1$。

(a)分别用数学归纳法和第 2 章例 19 的结论，证明：

(1) $F(n+2)=1+\sum_{i=0}^{n} F(n)$；　　　(2) $F(n+2)\ge\left(\frac{1+\sqrt{5}}{2}\right)^n$；

(b)设计算法根据递归方程计算 $F(n)$，将时间复杂度表达成 n 的函数;

(c)根据第 2 章例 19 中 $F(n)$的解析表达式，设计算法计算 $F(n)$，将时间复杂度复杂性表达成 n 的函数，并指明计算机运行该算法时可能遇到的问题。

(d)根据表达式 $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}F(n+1) & F(n)\\ F(n) & F(n-1)\end{pmatrix}=\begin{pmatrix}F(n+2) & F(n+1)\\ F(n+1) & F(n)\end{pmatrix}$ 和习题 3.1(b)的算法，设计算法计算 $F(n)$，将时间复杂度复杂性表达成 n 的函数。

(e)比较(b),(c),(d)得到的算法。

3.3 给定平面上 n 个点构成的集合 $S=\{p_1,\dots,p_n\}$。如果存在边平行于坐标轴的矩形仅包含 S 中的两个点 p_i 和 p_j（$1\le i,j\le n$）,则称 p_i 和 p_j 为“友谊点对”。试设计一个分

治算法统计 S 中友谊点对的个数。

3.4 给定平面上 n 个点构成的集合 S，试设计一个分治算法输出 S 的三个点，使得以这三个点为顶点的三角形的周长达到最小值。(提示：模仿最邻近点的分治过程)

3.5 证明求解凸包问题的蛮力算法的正确性。

3.6 给定平面上 n 个白点和 n 个黑点，试设计一个分治算法将每个白点与一个黑点相连，使得所有连线互不相交，分析算法的时间复杂度。（提示：划分时类似于 GrahamScan 算法考虑极角，确保子问题比较均匀）

3.7 给定凸多边形 $p_1,p_2,\ldots,p_n$（边界逆时针顺序）和 n 个点 $q_1,q_2,\ldots,q_n$，试设计一个分治算法计算 $q_1,q_2,\ldots,q_n$ 中位于凸多边形 $p_1,p_2,\ldots,p_n$ 内部的点的个数，使其时间复杂度是 n^2 的严格低阶函数。

3.8 输入含有 n 个顶点的加权树 T 和实数 τ，树 T 中每条边的权值均非负，树中顶点 x,y 的距离 $dis(x,y)$ 定义为从 x 到 y 的路径上各边权值之和。试设计一个分治算法输出满足 $dis(x,y)\leq\tau$ 的顶点对的个数。

3.9 设 $X[0:n-1]$ 和 $Y[0:n-1]$ 为两个数组，每个数组中的 n 个均已经排好序，试设计一个 O(log n)的算法，找出 X 和 Y 中 $2n$ 个数的中位数，并进行复杂性分析。

3.10 设 $A[1:n]$ 是由不同实数组成的数组，如果 $i< j$ 且 $A[i]>A[j]$，则称实数对 $(A[i], A[j])$ 是该数组的一个反序。如，若 A=[3,5,2,4]，则该数组存在 3 个反序(3,2)、(5,2)和(5,4)。反序的个数可以用来衡量一个数组的无序程度。设计一个分治算法（要求时间复杂度严格低于 n^2），计算给定数组的反序个数。

3.11 设单调递增有序数组 A 中的元素被循环右移了 k 个位置，如<35; 42; 5; 15; 27; 29>被循环右移两个位置（$k = 2$）得到<27; 29; 35; 42; 5; 15>。

(1). 假设 k 已知，给出一个时间复杂度为 $O(1)$ 的算法找出 A 中的最大元素。

(2). 假设 k 未知，设计一个时间复杂度为 O(log n) 的算法找出 A 中的最大元素。

3.12 设 M 是一个 $m\times n$ 的矩阵，其中每行的元素从左到右单增有序，每列的元素从上到下单增有序。给出一个分治算法计算出给定元素 x 在 M 中的位置或者表明 x 不在 M 中，分析算法的时间复杂性。

3.13 给定平面上 n 个白点和 n 个黑点，试设计一个算法判定是否存在一条不通过任何输入点的直线使得所有白点和所有黑点分别位于直线的两侧。（提示：对每个点利用划分寻找可能的直线，划分时类似于 GrahamScan 算法考虑极角）