2.1、2.2节，总共任选10题

2.1 排序问题

Problem 2.1 (冒泡排序(必选))

冒泡排序(算法2)对数组$A[1..n]$中的元素进行排序。

Algorithm 2: BUBBLE-SORT($A[1..n]$)

```
for i := n downto 2 do
    for j := 1 to i − 1 do
        if A[j] > A[j + 1] then
            SWAP(A[j], A[j + 1]) ;
```

1. 请证明冒泡排序的正确性；

2. 请分析冒泡排序的最坏情况、平均情况时间复杂度；(注：我们以元素的比较为关键操作，数组元素的交换不计入关键操作。)

3. 我们可以改进冒泡排序来避免在数组尾部进行的不必要的比较操作，具体做法是记录每次循环中最后发生元素交换的位置。假设最后一次元素交换发生在$A[k]$和$A[k+1]$之间，则我们记录下这一位置。未来的数组遍历就不用再扫描下标k之后的元素。请说明这样的做法是否会影响算法的最坏情况和平均情况时间复杂度。

Problem 2.2

请证明，假设一棵二叉树的高度为h，叶节点个数为L，则我们有：$L \leq 2^h$。

Problem 2.3

给定正整数数组$A = [93, 23, 8, 46, 26, 43, 6, 97]$。分别使用插入排序、快速排序、合并排序对该数组进行排序。请图示各个算法执行的详细过程。

Problem 2.4

1. 给出一个算法，它在最坏情况下可以只利用五次比较来对四个元素进行排序；

2. 给出一个算法，使得在最坏情况下，对五个元素进行排序时，该算法有最优的表现。

Problem 2.5

请修改QUICK-SORT算法中的PARTITION过程

1. 对数组中的n个整数重新排序，使得所有负数排在非负数的前面(对元素的大小关系没有要求)；

2. 对数组中的n个球进行排序，使得所有黄球在最左边，红球在中间，篮球在最右边(假设所有球只有这三种颜色)。

Problem 2.6 (The d-ary Heap (必选))
一个$d-ary$堆可以用一个一维数组表示。根节点存放在$A[1]$，它的子女节点依次放在$A[2],...,A[d+1]$，这些子女节点的子女节点存放在$A[d+2],...,A[d^2+d+1]$，依次类推。请证明下面两个分别计算节点下标为i的节点的父节点和它的第j（$1 \leq j \leq d$）个子女节点的过程的正确性。

- D-ARY-PARENT(i)

  return $\lfloor \frac{i-2}{d}+1 \rfloor$;

- D-ARY-CHILD(i, j)

  return d(i-1) + j +1;

Problem 2.7
有两个有序数组A与B，它们的长度分别为k，m，其中$k+m=n$，证明：将这两个数组进行合并时，合并的不同形式的个数为$\binom{n}{k}=\binom{n}{m}$。

Problem 2.8
假设两个待合并的数组长度分别为k与m，其中k远小于m。试给出一个合并算法，它能够利用k远小于m这个特点，使得其比较次数至多为$(k+m)/2$。在最坏情况下，为了达到上述上界，k必须多小？是否存在一个k的取值范围，使得比较次数的上界为$\sqrt{k+m}$？你能否得出在这一情况下元素的移动次数？

Problem 2.9
我们称一个数组$A[1..n]$为k-sorted，如果它能够被分为k段，每段长度为n/k，左边段中的元素一定小于右边段中的元素(每段之内的元素可以是乱序的)。请设计一个$O(n\log k)$的算法，将数组$A[1..n]$排成k-sorted。(算法分析过程中，可以假设n总是k的倍数，且n和k都是2 的幂。)

Problem 2.10 (匹配螺丝和螺母(必选))
给定n个螺钉，它们的尺寸互不相同，另外给定n个螺母，每个螺钉与唯一的螺母相匹配。你可以用一个螺钉旋入一个螺母来判断它们是否匹配，判断的结果有三种可能：正好匹配、螺母比螺钉大、螺母比螺钉小。你不可以将两个螺钉或者螺母直接进行比较。请给出一个平均情况时间复杂度为$O(n\log n)$的算法找出所有匹配的螺钉和螺母。

Problem 2.11 (计算逆序对的个数((必选)))
假定数组$A[1..n]$中存放着n个各不相同的整数。

1. 请设计一个$O(n\log n)$的算法计算给定数组中所有逆序对的个数；(提示：可以考虑将你的算法“附着”在合并排序之上。)
2. 我们可以将逆序对的定义进行推广：我们定义下标二元组(i,j)为一个广义逆序对，如果对于预先给定的正整数常数C，$i<j$，且$A[i]>C\cdot A[j]$。显然当$C=1$时，该定义退化为传统逆序对。请设计一个算法计算数组中所有的广义逆序对的个数。

2.2 分治策略

Problem 2.12 (常见项问题(必选))

给定数组$R[1\cdots n]$，唯一可以对它进行的操作是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。如果数组中至少有$\frac{n}{13}$个元素相等，那么该元素被称为“常见元素”。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元素。

1. 请设计一个分治算法找到所有的常见元素：
   i) 证明R中存在最多13个不同的常见元素；
   ii) 设x是$R[1\cdots n]$中的常见元素，证明x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元素。
   iii) 请根据上面的讨论设计一个分治算法。
2. 常见元素定义中的13换为任意大于等于2的正整数k，算法是否还正常工作？
3. 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$\frac{n}{2}$的元素)，是否有更高效的算法解决该问题？
4. 常见元素问题的难度(下界)是什么，它是否比比较排序更难或更容易？

Problem 2.13

你在一个有n个代表的政治会议会场内，每个代表都隶属于一个政党，但是并不知道他们属于哪个政党。假设你直接询问一个代表，他会拒绝回答，但是你可以通过介绍两个代表认识来分辨他们是否属于同一个政党(因为同一政党的代表会礼貌地握手并给予对方微笑；不同政党的代表会怒视对方)。

1. 假设代表中的大多数(一半以上)来自同一政党(称之为主要政党)。请设计一个算法来判定每个代表是否属于这个主要政党；
2. 假设代表们来自k个政党，一个政党占多数当且仅当属于它的代表的数目比其他任何政党的代表都多。请设计一个算法找出一个来自占多数的政党的代表，或者返回不存在占多数的政党。

Problem 2.14 (寻找maxima(必选))

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。

1. 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，你能否设计一个同样高效的分治算法？
2. 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$T(n)=O(n)$；

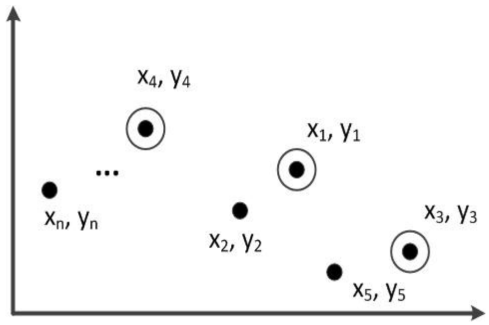

图 2.1: 平面上的maxima点

- 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；

3. 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 2.15 (整数乘法)
给定两个n个比特的整数，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。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：

1. 算法的最坏情况时间复杂度为$O(n^2)$；
2. 算法的最坏情况时间复杂度为$o(n^2)$。

Problem 2.16 (矩阵乘法)
给定两个大小为$n \times n$的矩阵，请给出两个分别符合下面要求的算法来计算它们的乘积。假设矩阵中均为整数，两个整数的相乘/相加为关键操作。

1. 算法的最坏情况时间复杂度为$O(n^3)$。
2. 算法的最坏情况时间复杂度为$o(n^3)$。

Problem 2.17 (距离最近的点对)
给定平面上的n个点，请在$O(n \log n)$的时间内找出距离最近的一对点。

Problem 2.18 (芯片测试)
一教授有n个VLSI芯片，有的是好芯片，有的是坏芯片。现在我们需要检测每个芯片的好坏。教授的测试装置一次可以将两枚芯片互连在一起作测试：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正确报告另一片的好坏，但一个坏的芯片的结果可能是正确的，也可能是错误的。

(a) 对于测试的四种可能结果，请证明下面的四个结论；

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

(b) 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。

(c) 假设有多于$n/2$个芯片是好的，请设计一个蛮力算法，判定所有芯片的好坏；

(d) 假设有多于$n/2$个芯片是好的，请设计一个分治算法，在$O(n)$的时间内判定每个芯片的好坏。