习题，可用 word 或者手写拍照上传

利用递归树来猜测递归式 $T(n)=3T(\lfloor n/2 \rfloor)+n$ 的一个好的渐近上界，并利用代换法来证明你的猜测。

利用递归树来证明递归式 $T(n)=T(n/3)+T(2n/3)+cn$ 的解是 $\Omega(n\lg n)$，其中 c 是一个常数。

用主方法来给出下列递归式紧确的渐近界：

a) $T(n)=4T(n/2)+n$

b) $T(n)=4T(n/2)+n^2$

c) $T(n)=4T(n/2)+n^3$

某个算法 A 的运行时间由递归式 $T(n)=7T(n/2)+n^2$ 表示；另一个算法 A' 的运行时间为 $T'(n)=aT'(n/4)+n^2$。若要 A' 比 A 更快，那么 a 的最大整数值是多少？

习题 2-3 改写二分搜索算法

设 a[0:$n-1$]是已排好序的数组。请改写二分搜索算法，使得当搜索元素 x 不在数组中时，返回小于 x 的最大元素位置 i 和大于 x 的最小元素位置 j。当搜索元素在数组中时，i 和 j 相同，均为 x 在数组中的位置。

习题 2-6 矩阵乘法

对任何非零偶数 n，总可以找到奇数 m 和正整数 k，使得 $n=m2^k$。为了求出 2 个 n 阶矩阵的乘积，可以把一个 n 阶矩阵分成 $m\times m$ 个子矩阵，每个子矩阵有 $2^k\times 2^k$ 个元素。当需要求 $2^k\times 2^k$ 的子矩阵的积时，使用 Strassen 算法。设计一个传统方法与 Strassen 算法相结合的矩阵相乘算法，对任何偶数 n，都可以求出 2 个 n 阶矩阵的乘积。并分析算法的计算时间复杂性。

习题 2-7 多项式乘积

设 $P(x)=a_0+a_1x+\cdots+a_dx^d$ 是一个 d 次多项式。假设已有一个算法能在 $O(i)$ 时间内计算一个 i 次多项式与一个一次多项式的乘积，以及一个算法能在 $O(i\log i)$ 时间内计算 2 个 i 次多项式的乘积。对于任意给定的 d 个整数 $n_1,n_2,\cdots,n_d$，用分治法设计一个有效算法，计算出满足 $P(n_1)=P(n_2)=\cdots=P(n_d)=0$ 且最高次项系数为 1 的 d 次多项式 $P(x)$，并分析算法的效率。

习题 2-28　X 和 Y 的中位数

设 X[0:$n-1$]和 Y[0:$n-1$]为 2 个数组，每个数组中含有 n 个已排好序的数。试设计一个 $O(\log n)$时间的算法，找出 X 和 Y 的 $2n$ 个数的中位数。

编程题，提交 LeetCode 截图

33. 搜索旋转排序数组 - 力扣（LeetCode）
50. Pow(x, n) - 力扣（LeetCode）
69. x 的平方根 - 力扣（LeetCode）
162. 寻找峰值 - 力扣（LeetCode）
215. 数组中的第 K 个最大元素 - 力扣（LeetCode）
540. 有序数组中的单一元素 - 力扣（LeetCode）