算法基础第一课

重新认识程序

程序 = 数据结构 + 算法

学习算法可以算作一种
学术研究

如果按照有用没用来指导学术研究，就会变成学术研究的灾难，
学术研究者只关注研究本身

时间复杂度

时间复杂度为，一个算法流程中，常数操作数量的指标，这个 指标叫做O，big O。具体为，如果常数操作数量的表达式中， **只要高阶项，不要低阶项，也不要高阶项系数之后**，剩下的部 分记为f(N)，那么该算法的时间复杂度为O(f(N))

复杂度O(N)

题目一：

求一个整型列表的元素的最大值

复杂度O(N^2)

题目二

对一个整型列表进行升序排序

复杂度O(logN)

题目三

查找**有序**列表中的指定元素，返回所在位置，如果找不到，返回-1

复杂度O(N+M)

题目四

对于两个有序列表A和B，长度分别是N和M，求两个数组的公共元素（**阿里2017**）

空间复杂度

空间复杂度，是对一个算法在运行过程中**临时**占用存储空间大小的量度，**额外空间，**记做S(n)=O(f(n))

题目五

给定一个列表 [1,2,3,4,5,6,7],希望得到新列表[6,7,1,2,3,4,5]

1. 申请临时列表进行逆序存储，空间复杂度：O(N)

2. 反转列表，O(1)

冒泡排序

时间复杂度O(N^2)，额外空间复杂度O(1)，可以做到稳定性

插入排序

时间复杂度O(N^2)，额外空间复杂度O(1)，可以做到稳定性

归并排序

时间复杂度O(N*log(N))，额外空间复杂度O(N)，可以做到稳定性

可以做到空间复杂度O(1),归并排序内部缓存法

Master公式

估计一个算法流程的复杂度，需要对流程的细节彻底知晓，但是对于递 归函数，有一个重要内容:这是一个估计递归行为复杂度的公式，但是 要求递归行为中，每次递归的规模是固定的。

如何使用master公式?

T(N) = aT(N/b) + N^d

如果

1，log(b,a) > d -> T(N) 的复杂度为 N^(log(b,a))

2，log(b,a) == d -> T(N) 的复杂度为 N^d * logN

3，log(b,a) < d -> T(N) 的复杂度为 N^d

归并排序的应用(进阶)

1. 小和问题

例如，数组s=[1,3,5,2,4,6]，在s[0]的左边小于等于s[0]的数的和为0，在s[1]的左边小于或等于s[1]的数和为1，在s[2]的左边小于等于s[2]的数和为1+3=4…以次类推，所以s的小和为0+1+4+1+6+15=27，
给定一个数组，实现函数返回小和

2. 逆序对问题（算法导论2-4）

在数组中的两个数字如果前面一个数字大于后面的数字，则这两个数字组成一个逆序对。给你一个数组，求出这个数组中逆序对的总数。
概括：如果a[i] > a[j] 且 i < j， a[i] 和 a[j] 构成一个逆序对。如：序列 [2, 4, 1, 3, 5] 中，有 3 个逆序对 (2, 1), (4, 1), (4, 3)，则返回 3 。