# 第一章 绪论

数据类型从低到高自动转换(int - float)

逻辑结构：数据元素之间的逻辑关系，即结构中定义的‘关系’

线性表：

1. 线性表的元素个数是有限的
2. 线性开始节点没有前驱
3. 线性结束节点没有后继

算法：

时间

1. 复杂度 （执行时间效率）
2. 空间复杂度 （占用存储空间)

例子：

1. 水仙花数

所谓水仙花数：指一个3位数，其各位数字的立方和等于数字本身，例如：153=1\*\*\*3+5\*\*\*3+3\*\*\*3

求出100~999之间的水仙花数

算法：比如 i

百位 a (1) a=i//100 (python //整除 其他语言/整除)

A = int(i /100)

十位 b (1) b=(i%100)//10

B = int((i-100\*a)/10)

个位 c (1) c=i%10

C = i-a\*100-b\*10

C = i-int(i/10)\*10

1. 计算10000以内的自守数

自守数：某个数的平方的末尾数等于这个数

Eg: 5x5=25

6x6 =36

25x25=625

76x76=5776

找规律：3位自守数是625和376，四位自守数是9376，5位自守数是90625

可以得知,n+1位自守数出自n位自守数

如果知道n位自守数为a，那么n+1的自守数应该在a的前面再加一个数

算法：n\*n%(10\*\*len(str(n)))) ==n

1. 出售金鱼

鱼商A将养的一缸金鱼分5次出售，第一次卖出全部的1/2加1/2条，

第二次卖出余下的1/3加1/3条，第三次卖出余下的1/4加1/4条，

第四次卖出余下的1/5加1/5条,最后卖出余下的11条.问原来鱼缸多少条鱼?

算法:第j次卖余下的1/(j+1)加/(j+1)条

假设第j次鱼的总数为x条

第j次留下（x-(x+1)/(j+1)）条

第一次x x -1/2\*x-1/2

第二次x -1/2\*x-1/2 x -1/2\*x-1/2-(1/3x-1/3)

第三次x -1/2\*x-1/2-(1/3x-1/3) x -1/2\*x-1/2-(1/3x-1/3)-(1/3(x-1/2\*x-1/2-(1/3x-1/3) +1/3)

第四次x -1/2\*x-1/2-(1/3x-1/3)-(1/3(x-1/2\*x-1/2-(1/3x-1/3) +1/3) x -1/2\*x-1/2-(1/3x-1/3)-(1/3(x-1/2\*x-1/2-(1/3x-1/3)+1/3)-1/4(x-1/2\*x-1/2-(1/3x-1/3)-(1/3(x-1/2\*x-1/2-(1/3x-1/3) +1/3)+1/4)

递归算法：

59

第一次卖出30， 剩余 29

第二次卖出10， 剩余19

第三次卖出5， 剩余14

第四次卖出3， 剩余11

小明有5本书，借给甲乙丙三位小朋友，每人每次只能借一本，有多少种不同借法

A,b,c,d,e,f

(a,b,c) (a,b,d)(a,b,e)(a,b,f)

某天夜里，a,b,c,d,e,五人一起去捕鱼，到第二天凌晨时都疲惫不堪，于是各自找地方睡觉，天亮啦，a第一个醒来，他将鱼分为5份，把多余的一条鱼扔掉，拿走自己的一份，

b第二个醒来，他将鱼分为5份，把多余的一条鱼扔掉，拿走自己的一份，c,d,e依次醒来，按照同样方法拿走鱼，问他们伙伴至少捕捉了多少条鱼？ 3121