# 01

算法：

输入：算法具有零个或多个输入

输出：算法至少由一个或多个输出

有穷性

确定性

可行性

时间复杂度分析：

执行次数==时间

用大写O（）来体现算法时间复杂度的记法，我们称之为大O记法

推导大O阶方法攻略：

——用常数1取代运行时间中的所有加法常数。

——在修改后的运行次数函数中，只保留最高阶项

——如果最高阶项存在且不是1，则去除于这个项相乘的常数

——得到的最后结果就是大O阶

由2^x=n得到x=log（2）n，所以这个循环的时间复杂度为O（logn）//由于log以不同的底数为底时，成倍数关系，因此可以把底数去掉。

常用的时间复杂度所耗费的时间从小到大一次是：

O（1）< O（logn）<O(n)<O(nlogn)<O(n^2)<O(n^3)<O(2^n)<O(n!)<O(n^n)

# 02

线性表元素的个数n（n>=0）定义为线性表的长度，当n=0时，，称为空表，

抽象数据类型：抽象数据类型的定义仅仅取决于它的一组逻辑特性，而与它在计算机中的表示和实现无关。

ADT抽象数据类型名

Data数据元素之间逻辑关系的定义

Operation操作

endADT

ADT线性表（list）//比较适合元素个数比较难稳定，不经常插入和删除元素，而更多的操作是存取数据的应用。

Data

线性表的数据对象集合为{a1,a2,...,an}每个元素的类型均为DataType。其中，除第一个元素a1外，每一个元素有且只有一个直接前驱元素，除了最后一个元素an外，每一个元素有且之只有一个直接后继元素。数据元素之间的关系时一对一的关系。

Operation

Initlist（\*L）：初始化操作，建立有个空的线性表L。

ListEmpty（L）：判断线性表是否为空表，若线性表为空，返回true，否则返回flase

ClearList（\*L）:将线性表清空。

GetElem（L，i，\*e）：将线性表L中的第i个位置元素值返回给e。//按照从1开始，数组从0开始

LocateElem（L，e）：在线性表L中查找给定值e相等的元素，如果查找成功，返回该元素在表中序号表示陈工。否则，返回0表示失败。//这里给人看，一就是一

ListInsert（\*L，i，e）：在线性表L中第i个位置插图新元素e。

ListDelete（\*L，i，\*e）：删除线性表L中第i个位置元素，并用e返回其值。

ListLength（L）：返回线性表L的元素个数。

EndADT

线性表顺序存储的结构代码

#define MAXSIZE 20

Typedef int Elemtype；

Typedef struct

{

Elemtype data[MAXSIZE];

Int length;//线性表当前长度

}SqList；

、

LOC表示获得存储位置的函数

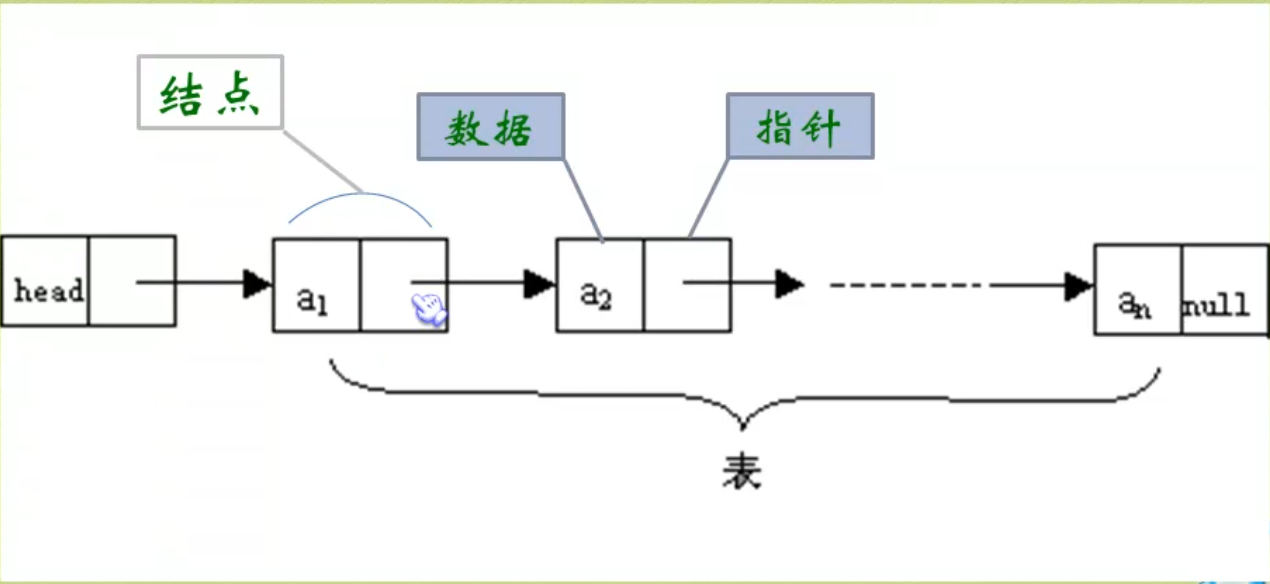
Eg：c便是存储单元

LOC(ai+1)=LOC（ai）+c

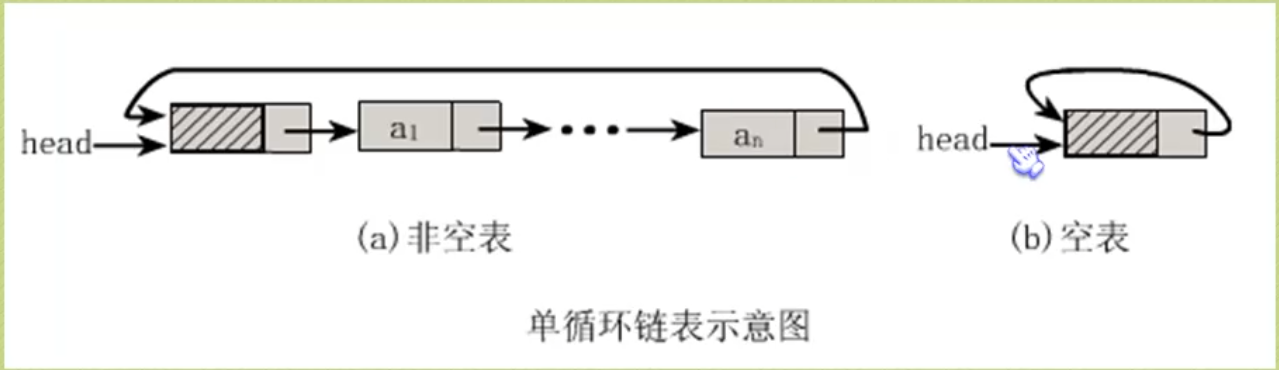
LOC（ai）=LOC(ai)+（i-1）\*c//

返回值类型Status时一个整型，约定返回1代表OK，返回0代表Error.

# 03







循环链表中：终端结点用尾指针rear指示。

指针之间用->表示地址；元素之间用“.”表示。